ZIV-ZAKAI BOUND APPLIED TO AN AMPLITUDE-COMPARISON MONOPULSE RADAR

THESIS

Terry E. Smith
Second Lieutenant, USAF

AFIT/GE/ENG/84D-[illegible]

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/84D-62

ZIV-ZAKAI BOUND APPLIED TO AN AMPLITUDE-COMPARISON MONOPULSE RADAR

THESIS

Terry E. Smith
Second Lieutenant, USAF

AFIT/GE/ENG/84D-62

Approved for public release; distribution unlimited

AFIT/GE/ENG/84D-62

ZIV-ZAKAI BOUND APPLIED TO AN AMPLITUDE-COMPARISON MONOPULSE RADAR

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Terry E. Smith, B.S.

Second Lieutenent, USAF

December 1984

Acknowledgements

I am, most of all, grateful to my family for providing me the encouragement and time necessary to complete this thesis. They deserve recognition for the many hours of neglect they endured. Additionally, I wish to express my appreciation to Lt. Col Carpinella, my advisor, for suggesting this thesis topic and guidance throughout the development. His enthusiastic attitude and expertise in radar signal processing were invaluable in this learning experience. Finally, I would like to thank Major Castor and Dr. Pyati, my readers, for their constructive criticisms of my work.

Table of Contents

List of Figures

Conventions

The symbol | | means the magnitude of the terms contained within

The symbol Re{ } means the "real part" & Im{ } means "imaginary part"

The symbol <----> indicates a Fourier transform pair

The symbol $\propto$ refers to "proportional to"

The use of $\hat{}$ over a symbol means the estimate of an unknown parameter

Random variables are uppercase; random variable values are lowercase

A / within parantheses {/A} means "given that" {"given that" A is true}

The symbol ln[ ] denotes the natural logarithm of the enclosed quantity

The symbol sqrt( ) means the square root operation

The symbol E{ } means the expected value

The symbol var{ } means the variance

The symbol cov{ } means the covariance

The symbol P{ } means the probability of the enclosed random quantity

Boldface letters denote vector quantities

$\sum$ refers to a sum & $\int$ refers to an integration

A'( ) refers to the derivative with respect to the parameter A

A"( ) refers to the second derivitive with respect to the parameter A

wrt means "with respect to"

E, or Egy, designates signal energy

BW designates the 3dB beamwidth

PRI designates the pulse repetition interval

MSE designates mean-square error

CRB designates the Cramer-Rao Bound

ZZB designates the Ziv-Zakai Bound

ABSTRACT

This investigation addresses the problem of estimating the target angle off beam-axis of an amplitude-comparison monopulse radar to generate an error signal that can be applied to a servo-control system to reposition the beam-axis on target, thus providing target tracking. Cramer-Rao (CR) and Ziv-Zakai (ZZ) bounds are derived to indicate system performance under varying signal-to-noise ratio (SNR) conditions. Actual tracking error is approximated from a computer simulated tracking loop and then compared to the CR and ZZ bounds for varying SNR levels, for tracking in "slow" and "rapid" target fluctuation environments, and for tracking in the presence of specular and diffuse multipath. A two-dimensional tracking model and associated ZZ performance bound are also presented.

At high SNR levels, the CR bound results lower bounded the mean-square tracking error, but for low SNR conditions the CR bound exceeded the mean-square error. The ZZ bound results indicate a tight lower bound for the mean-square tracking error at low SNR levels and in both the target fluctuation and multipath environments. "Slow" target fluctuations and diffuse multipath results indicate that target tracking capability is not seriously degraded in either of these two surroundings. Conversely, "Rapid" target fluctuations and specular multipath environment results indicate serious tracking degradation is introduced in the amplitude-comparison monopulse tracker.

ZIV-ZAKAI BOUND APPLIED TO AN AMPLITUDE-COMPARISON MONOPULSE RADAR

I. INTRODUCTION

Background

An amplitude-comparison monopulse radar is a tracking radar that determines target direction by comparing the reflected signal amplitudes received simultaneously on two identical but noncoincident antenna patterns. Target tracking is accomplished by maintaining the axis of symmetry of the antenna patterns (beam-axis) on the target. As the target moves off beam-axis, the amplitude-comparison yields an error voltage proportional to the angular difference between the present beam-axis location and the target position. This error information is applied to a servo-control system to reposition the beam-axis on target, thus providing target tracking.

Due to practical considerations such as channel noise and servo time-constant limitations, the radar beam-axis will be steered in the vicinity of the actual target position. Associated with this vicinity is the tracking error from the true target location. An accurate bound on the tracking error provides a direct indication of the quality of the estimated target position and thus provides a measure of the ability of the system to track a target.

Traditionaly, the tracking error has been lower bounded by the application of the Cramer-Rao bound to the variance of the distribution

of the estimated angle off beam-axis. This bound limits the variance of the estimate to a value directly proportional to the radar's beamwidth (BW) and inversely proportional to the available signal-to-noise ratio (SNR).

$$\text{var(estimated position)} \propto \frac{BW}{SNR}$$

Inspection of the Cramer-Rao bound indicates that for any signal-to-noise ratio, (SNR), reducing the beamwidth decreases the variance of the estimate. This presents a shortcoming of the Cramer-Rao bound as this relationship will not hold for all BW and SNR values. Thus, SNR levels for which the Cramer-Rao bound holds (ie the variance of the estimated position is directly proportional to the beamwidth) must be defined if the bound is to be useful.

Problem and Scope

This thesis concentrates upon generating a maximum-likelihood estimate of the target angle off beam-axis, determining the mean-square error between the estimate and the actual target position, and applying the Ziv-Zakai bound to verify system performance.

The maximum-likelihood estimate of an unknown parameter is the parameter value that would most likely cause a given observation to occur. In terms of statistical information, the maximum-likelihood estimate would be the value of an unknown parameter for which the conditional probability distribution function, given the unknown parameter, is maximum. This thesis develops a maximum-likelihood estimate of the target angle off beam-axis and uses the estimate to

track the target.

The optimum solution of the estimation of an unknown quantity from data is, in general, very difficult analytically. The solution, however, can be considerably simplified if the permitted analytical operations are optimum in the mean-square sense. The error of the target position in the mean-square sense is the average value of the squared difference between the estimated and actual target position:

$$\text{mean-square error} = E\{(\text{estimated position} - \text{actual position})^2\}$$

This thesis considers the difference between the actual target position and the maximum-likelihood estimated position in the mean-square sense.

The Ziv-Zakai bound is designed to be tighter to actual performance values at low signal-to-noise ratio levels. The bounds established by the Ziv-Zakai method are derived by comparing the estimation problem with the known results from detection theory. The derived bound compares a suboptimal estimate of one of two possible values of a parameter to an optimal detection scheme between the two possible values. A criteria for determination of which unique parameter value was estimated then leads to a simple comparison between the probability of error associated with optimal and suboptimal detection methods. A form of the Ziv-Zakai bound is used in this thesis to derive a lower bound for the mean-square tracking error.

The goal of this thesis is to solve the general problem of estimating an amplitude-comparison monopulse radar target parameter and to apply the Ziv-Zakai bound to develop an accuracy estimate for system performance as a function of signal-to-noise ratio.

Assumptions

The development and solution of this thesis depends upon the following assumptions:

1) The channel noise is assumed to be additive white Gaussian that is independent from channel to channel and pulse to pulse with zero mean and variance equal to $N_0/2$.
2) The model is restricted to one dimensional tracking; thus total independence from a second dimension is assumed.
3) Any velocity changes are assumed to be slowly varying with respect to the time between radar pulses.
4) The antenna patterns are derived assuming a uniform aperture illumination and the target is in the "far-field".
5) The amplitude of the return signal is assumed either unknown, but non-random, (deterministic) or a Rayleigh distributed random variable with completely specified characteristics.
6) The carrier phase, although unknown, is assumed to be the same in all receiver beams.

General Approach

The approach to solving this problem can be broken up into the following main divisions:

1) Develop a pair of observations that are weighted functions of the signals received by the two antenna patterns.

2) Process the observations to quantify a maximum-likelihood angle error estimate that will be used to steer the radar beam-axis to the predicted target location.

3) Derive Cramer-Rao and Ziv-Zakai bounds to indicate system performance under varying signal-to-noise ratio conditions.

4) Design and implement a computer simulated tracking loop that provides for beam-axis position changes and will yield the error between actual target position and the beam-axis position.

5) Convert the tracking error into a mean-square error and plot for different signal-to-noise ratio levels.

6) Compare the mean-square error plots to the derived Cramer-Rao and Ziv-Zakai bounds to indicate which bound provides the best estimate of accuracy, in the mean-square sense, for different signal-to-noise ratio levels.

7) Observe radar performance under situations of slow and rapid fading of the signal amplitude, and in the presence of a return embedded in multipath.

II. RADAR MODEL

The Antenna Functions

Amplitude-comparison monopulse radar uses two overlapping antenna patterns to obtain ang'e tracking for a single coordinate direction. The patterns are usually mirror images about the boresight axis (beam-axis). When a target is on the boresight axis, the signals received from the two patterns are equal. Target tracking can be accomplished by bringing the boresight axis into coincidence with the target bearing, as indicated by equal received signals.

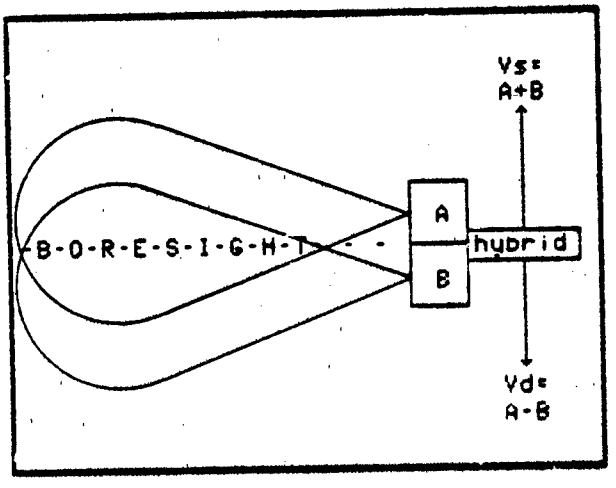

Figure 1. Simple Block Diagram of Antenna Circu...

Processing of the received signals is accomplished by applying the antenna patterns, A and B, to a hybrid junctio.. which sums and subtracts the two patterns. A block diagram of the amplitude-comparison antenna circuitry is shown in Figure 1.

The sum of the two beam patterns, A + B, is represented as an antenna function V_s. Likewise, the difference, A - B, is represented as V_d. To derive the two antenna functions, V_s and V_d, the following steps will be taken:

1) The technique for taking the Fourier transform of a uniform source will be developed.
2) A general expression will be derived for the magnitude of the electric field at a distance far from the source (far-field).
3) An expression will be developed for squinting (offsetting) the antenna patterns from the boresight axis.
4) Two antenna patterns will be combined with one squinted in a positive direction, and the other squinted an equal amount in the negative direction. These two patterns will then be added and subtracted to form V_s and V_d, respectively.
5) Approximations will be applied to the general expressions of 4 to develop simple expressions for the antenna functions.

Antenna Patterns

Each individual antenna pattern, A or B, is assumed to be generated from a uniformly illuminated rectangular aperture. A uniform rectangular aperture has aperture fields which are uniform in phase and amplitude across the physical aperture. The described situation for uniform illumination in the x-y plane is shown in Figure 2.

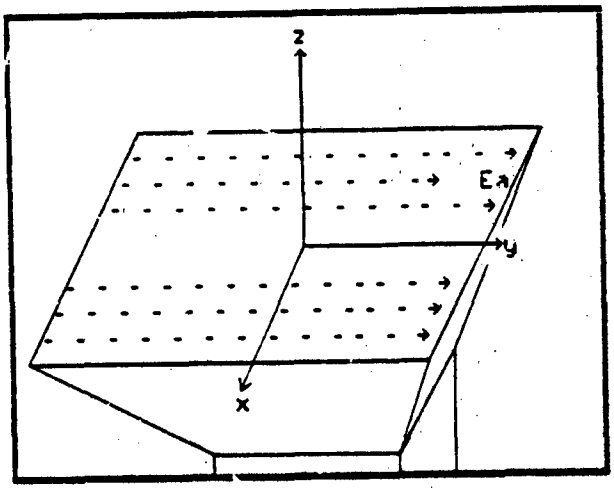

Figure 2. Uniformly Illuminated Rectangular Aperture in the x-y Plane

To specify antenna patterns, Fourier transform techniques are used. In general, the Fourier transform is used to convert from time domain to frequency domain. As applied to antennas, the Fourier transform can be used to convert from direct space (aperture illumination) to K-space (antenna far-field pattern) [1:523]. Three Fourier transform tools will be utilized in the far-field pattern development. These tools are:

1) $f(t) \longleftrightarrow F(u)$

2) $f'(t) \longleftrightarrow juF(u)$

3) $f(t) = \delta(t+a) \longleftrightarrow F(u) = \exp(jua)$

where:

f( ) refers to the illumination of the source

F( ) refers to the far-field antenna pattern

<----> designates the Fourier transform operation

primed variables refer to derivatives of that variable

δ( ) refers to the Dirac delta function [2:273]

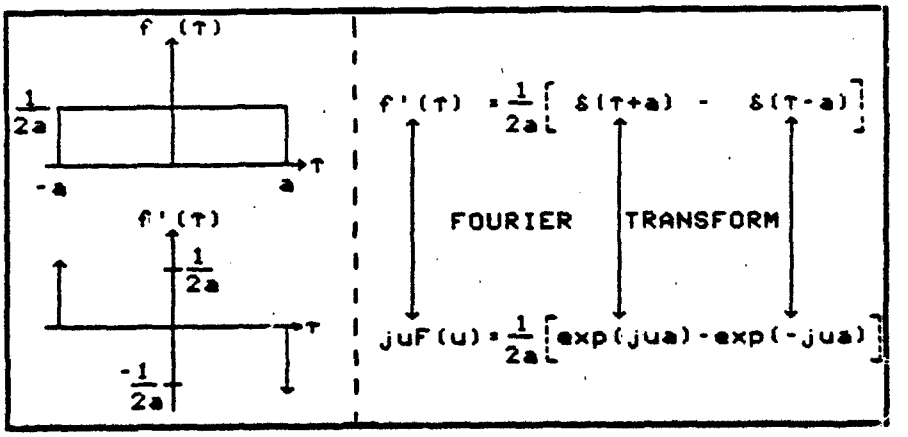

Figure 3. Graphical Determination of f'(t)

Consider f(t) to be a uniform source of length 2a, the derivative, f'(t), can be determined graphically as shown in Figure 3. Using the Fourier transform tools previously mentioned, the derivative of the uniform source can be utilized to determine the K-space representation of the far-field pattern.

$$f'(t) \longleftrightarrow juF(u) = (2a)^{-1}[\exp(jua) - \exp(-jua)]$$

Solving for F(u) yields:

$$F(u) = (2aju)^{-1}[\exp(jua) - \exp(-jua)]$$

$$= (2aju)^{-1}[2j\sin(ua)]$$

$$= (ua)^{-1}\sin(ua) = Sa(ua) \qquad (1)$$

where:

Sa( ) = sampling function [3:24]

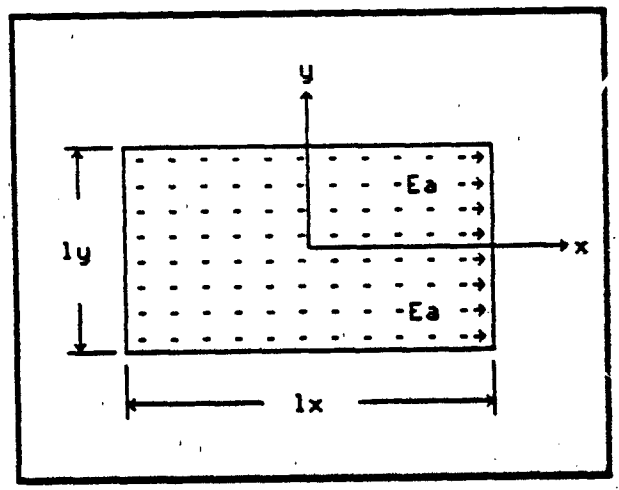

Figure 4. The Rectangular Aperture

To apply the Fourier expression for the far-field pattern, suppose the aperture electric field, E_a, is known and x directed as shown in the l_x by l_y rectangular aperture of Figure 4. For the one-dimensional problem, the source geometry takes the form of a uniform line source along the x-axis. The pattern of concern is the electric field in the x-z plane that will be generated by this uniform source. For the x directed source, the variable u will deacribe the x component of the phase constant for a plane wave. This phase constant is commonly refered to as k and equals 2(pi)/wavelength. Making a simple rectangular to spherical coordinates conversion, k_x = [sin(θ)cos(ϕ)]2(pi)/wavelength. Substituting this value into equation 1, and noting that the distance a = $l_x/2$, yields an expression for the magnitude of the electric field due to a uniform source.

$$E = Sa(2(pi)/wavelength)[sin(\theta)cos(\phi)]l_x/2 \qquad (1A)$$

For the one-dimensional radar model, further simplification of equation 1A can be made by noting that only the magnitude of the electric field in the x-z plane is of interest (ie electric field changes as a function of azimuth only). In the x-z plane:

$\phi = 0$

$\cos(\phi) = 1$

Simplifying equation 1A:

$$E = Sa[((pi)l_x/wavelength)\sin(\theta)] \quad (2)$$

For large apertures (ie l_x >> wavelength) the sin(θ) factor is approximately equal to θ [1:387]. Further, the factor $(pi)l_x/wavelength$ can be expressed as [4:267]:

$$(pi)l_x/wavelength = 2.78/BW \quad (3)$$

where:

BW = 3dB beamwidth

Substituting equation 3 into equation 2, a final form for the normalized magnitude of the electric field for the uniform rectangular aperture can be expressed as:

$$E = Sa(2.78\theta/BW) \quad (4)$$

A plot of equation 4 for varying θ/BW ratios is provided in Figure 5.

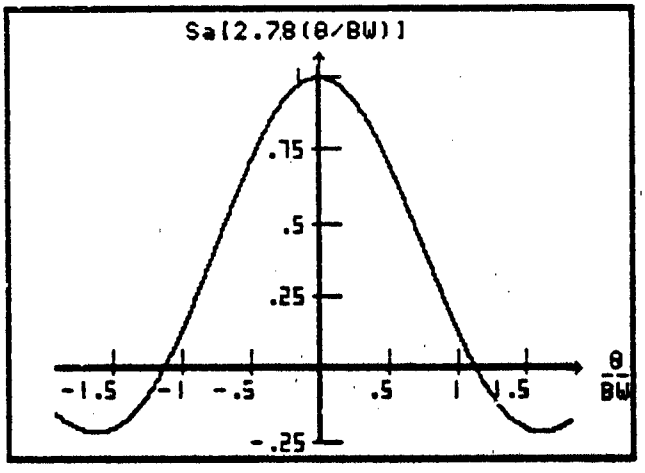

Figure 5. Antenna Pattern for Uniformly Illuminated Line Source

Squinting Beams Off Boresight

Amplitude-comparison monopulse uses two overlapping patterns, of the nature of equation 4, that are squinted (offset) from the boresight (tracking) axis. The boresight axis is an imaginary line drawn perpendicular, and centered on, the plane of the rectangular apertures. This squint relationship is shown in Figure 6, where one beam is squinted in a positive direction from boresight by the angle θ_s, and the other beam is squinted negatively from boresight by θ_s.

The squinting of the two beams from boresight will cause two distinct antenna voltage gains to be received from a target that is arbitrarily placed at some angle θ_t.

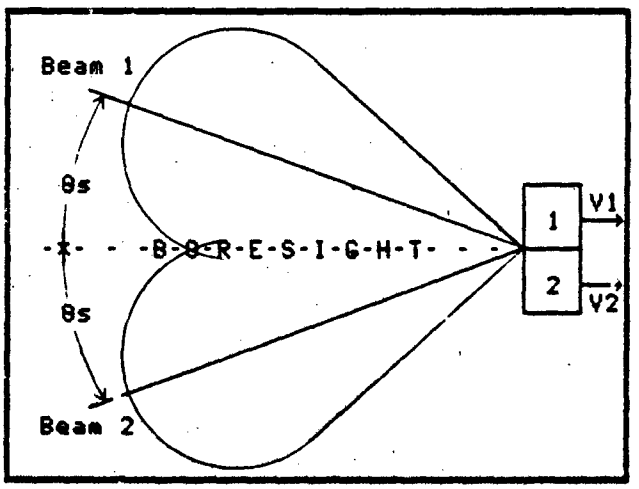

Figure 6. Squinting of Beams

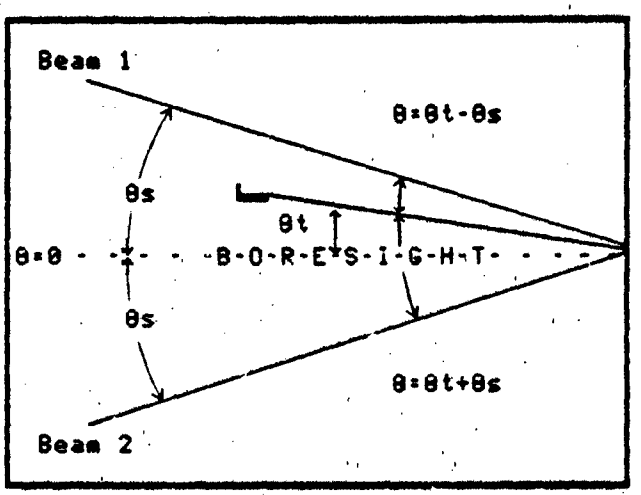

Figure 7. Geometry of Target Location with Squinted Beams

Figure 7 depicts the geometry of the described situation. Beam number one would locate the target at an angle $\theta_t - \theta_s$. Similarly, beam number two locates the target at an angle of $\theta_t + \theta_s$.

In order to locate a target position relative to the boresight axis, the monopulse radar sums and subtracts the voltage gain from each of the two squinted beams. The antenna functions V_s (sum of the patterns) and V_d (difference of the two patterns) depend upon completely specifying the squinted beam patterns. For a target located at an angle θ:

$$V_1(\theta) = E(\theta - \theta_s) \quad (5)$$

$$V_2(\theta) = E(\theta + \theta_s) \quad (6)$$

where:

$V_1(\theta)$ = received voltage due to positively squinted beam and is a function of the target angle θ

$V_2(\theta)$ = received voltage due to negatively squinted beam and is a function of the target angle θ

Using equations 5 and 6 with the patterns described by equation 4:

$$V_1(\theta) = Sa((2.78/BW)(\theta - \theta_s)) \quad (5A)$$

$$V_2(\theta) = Sa((2.78/BW)(\theta + \theta_s)) \quad (6A)$$

At this point of the antenna function development, the squint angle is established as one-half the beamwidth. With this assumption, equations 5A and 6A become:

$$V_1(\theta) = Sa(2.780/BW - 1.39) \quad (5B)$$

$$V_2(\theta) = Sa(2.780/BW + 1.39) \quad (6B)$$

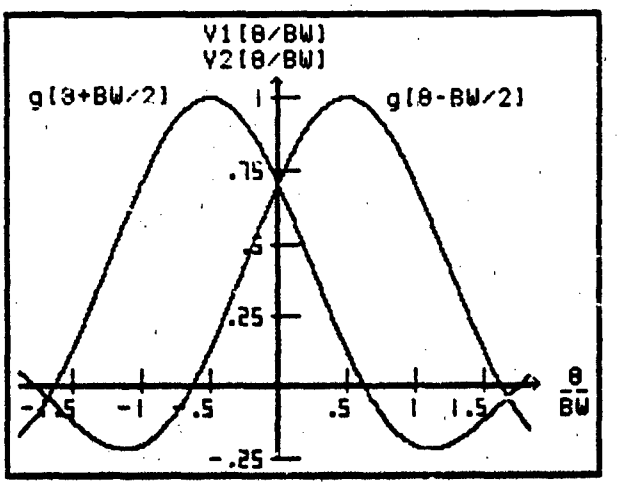

Figure 8. Voltage Gain of Two Beams Squinted Off Boresight by BW/2

A combined plot of equations 5B and 6B are shown in Figure 8. The antenna functions then follow as $V_s(\theta) = V_1(\theta) + V_2(\theta)$ and $V_d(\theta) = V_1(\theta) - V_2(\theta)$. The development for the sum pattern is as follows:

$$V_s(\theta) = V_1(\theta) + V_2(\theta)$$

$$= Sa(2.780/BW - 1.39) + Sa(2.780/BW + 1.39)$$

$$= Sa(2.780/BW - 1.39)\left[\frac{[2.780/BW + 1.39]}{[2.780/BW + 1.39]}\right]$$

$$+ Sa(2.780/BW + 1.39)\left[\frac{[2.780/BW - 1.39]}{[2.780/BW - 1.39]}\right]$$

let $a = 1.39$; $b = 2.780/BW$

$$V_s(\theta) = (a^2 - b^2)^{-1}\begin{bmatrix}[\sin(a - b) - \sin(a + b)]a + \\ [\sin(a - b) + \sin(a + b)]b\end{bmatrix}$$

$$= (a^2 - b^2)^{-1}\begin{bmatrix}[-\cos(a)\sin(b) - \cos(a)\sin(b)]b + \\ [\sin(a)\cos(b) + \sin(a)\cos(b)]a\end{bmatrix}$$

$$= (a^2 - b^2)^{-1}[2a\sin(a)\cos(b) - 2b\cos(a)\sin(b)]$$

$$V_s(\theta) = \frac{2}{7.73[\theta/BW]^2 - 1.93}[.5(\theta/BW)\sin(2.78\theta/BW) - 1.37\cos(2.78\theta/BW)] \quad (7)$$

A similar development for the difference pattern yields:

$$V_d(\theta) = \frac{2}{7.73[\theta/BW]^2 - 1.93}[.25\sin(2.78\theta/BW) - 2.74(\theta/BW)\cos(2.78\theta/BW)] \quad (8)$$

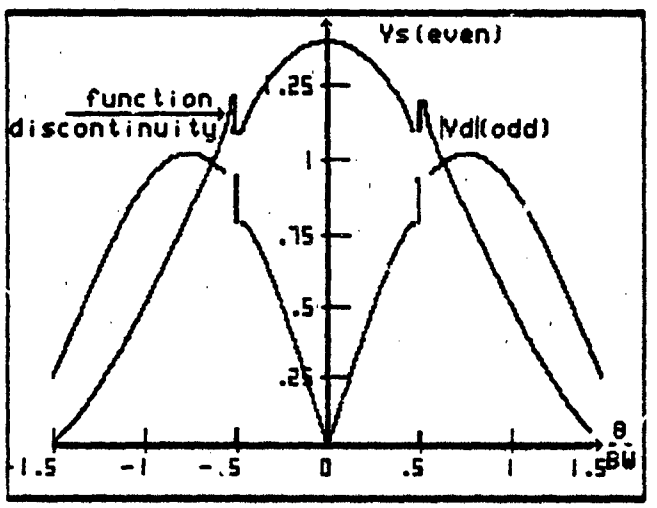

Figure 9. Monopulse Sum and Difference Patterns

In order to make subsequent calculations more tractable, and to remove the function discontinuities shown in Figure 9, equations 7 and 8 need to be simplified. Observation of Figure 9 indicates that V_s is an even function and V_d is an odd function. Consider approximating the sum pattern as a scaled cosine function and approximating the difference pattern as a scaled sine function:

$$V_s(\text{approx.}) = \text{sqrt}(2)\cos^n(c) \quad (7A)$$

$$V_d(\text{approx.}) = \sin^r(d) \quad (8A)$$

A few calculations on the arguments and powers of the approximations indicates that the best argument for the sum approximation is c = 1.047θ/BW, and the best power for the sum approximation sets n = 2; similarly, the best argument for the difference approximation is d = 1.88θ/BW, and the best power for the difference approximation sets n = 1. Substitution into equations 7A and 8A yield the final form for the antenna functions of the monopulse radar. These functions are:

$$V_s(\theta) = \text{sqrt}(2)\cos^2(1.047\theta/BW)$$

$$= (\text{sqrt}(2)/2)[1 + \cos(2.094\theta/BW)] \quad (9)$$

$$V_d(\theta) = \sin(1.88\theta/BW) \quad (10)$$

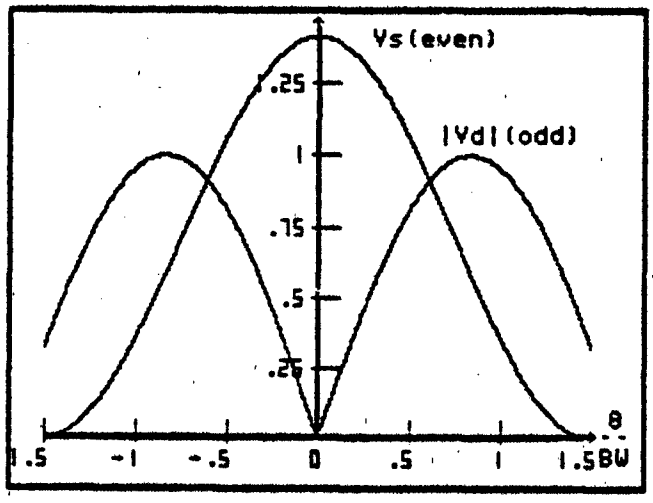

Figure 10. Approximations for Monopulse Antenna Patterns

For notational convenience, the theta dependence in the subsequent developments using the sum and difference patterns will be dropped and the antenna functions will be referred to as:

$$V_s(\theta) = V_s; \quad V_d(\theta) = V_d$$

Estimation Model

The general framework of the estimation problem consists of determination (estimation) of the value of an unknown parameter, possibly random, from an observation. Normally observations are made through a noisy channel, and in this case, estimation techniques must be used to determine the desired parameter value. Parameter estimation is then a mapping of the observation space into a parameter space.

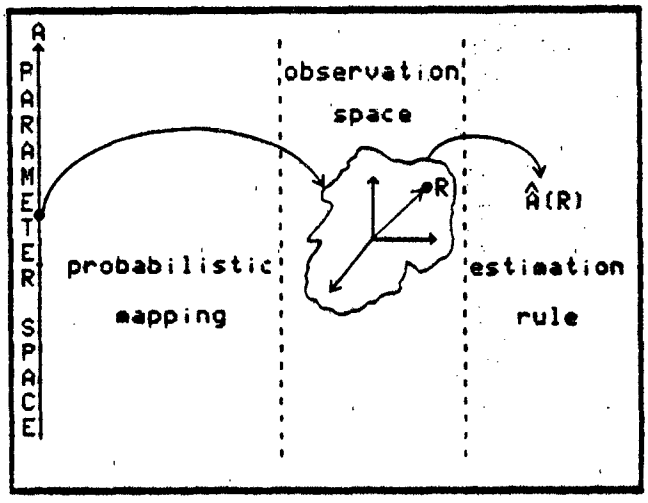

Figure 11. Estimation Model
Source: [5:53]

Figure 11 shows the basic components of the estimation model. Once a suitable rule for mapping into a parameter space has been developed, it is of interest to examine measures of quality of the estimation procedures. Possible quality measures might include the mean and the error (in the mean square sense) of the estimate. Frequently, quality measures of the estimation procedure are quite difficult, and for many cases rather than approach the error of the estimate directly it is of

more concern to derive a lower bound on the variance of any estimate. A comparison of the chosen estimation procedure to the derived lower bound will provide the needed measure of quality of the estimation procedure. To complete the estimation model for the amplitude-comparison monopulse radar, the following steps will be taken:

1) The observations that the estimator will have to work with are derived from the amplitude weighting provided by the antenna functions in the presence of additive white Gaussian noise.
2) A maximum-likelihood estimation procedure will be applied to the received observations to estimate the amplitude that most likely caused the observation. Subsequently, the amplitude estimate will be used to develop an error equation based upon the angle of arrival of the received signal from a target.
3) The Cramer-Rao lower bound will be developed which considers the conditional variance of any unbiased estimate for the target angle of arrival.
4) An alternate bound, the Ziv-Zakai bound, will be developed for the specific case where the procedure of estimation is maximum-likelihood.

Observation Formulation

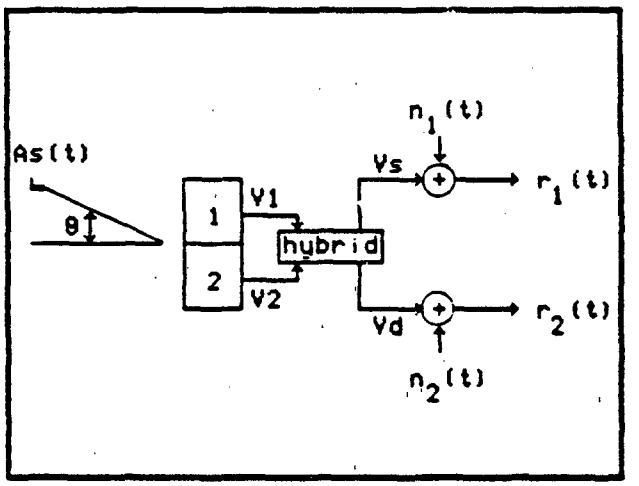

Figure 12. Monopulse Observation Model

For the specific case of additive white noise (See Figure 12) the received waveforms are:

$$r_1(t) = As(t)V_s + n_1(t) \quad (11)$$

$$r_2(t) = As(t)V_d + n_2(t) \quad (12)$$

$$r(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix}$$

where:

A = unknown signal amplitude (nonrandom)

s(t) = transmited signal

V_s = sum voltage antenna gain = $V_s(\theta)$

V_d = difference voltage antenna gain = $V_d(\theta)$

n(t) = additive white Gaussian noise

In general, the transmitted signal (s(t)); the disturbing noise

(n(t)); and the received signal are random processes. The key to analyzing the random observation is to find a way to replace all waveforms by finite dimensional vectors, for which characterization of the random process, by way of a joint density function, can be made. One implementation of this transformation is referred to as Gram-Schmidt orthogonalization [6:266]. The orthogonalization procedure permits the representation of any K finite energy time functions as linear combinations of $M \leq K$ orthonormal basis functions.

$$s_k(t) = \sum_{1}^{M} (s_{k,m}) \phi_m(t) \quad k=1,2,\ldots K \tag{13}$$

where:

$$s_{k,m} = \int s_k(t) \phi_m(t) dt$$

integration is taken over a period

$\{\phi_1(t), \phi_2(t), \ldots \phi_M(t)\}$ are orthonormal

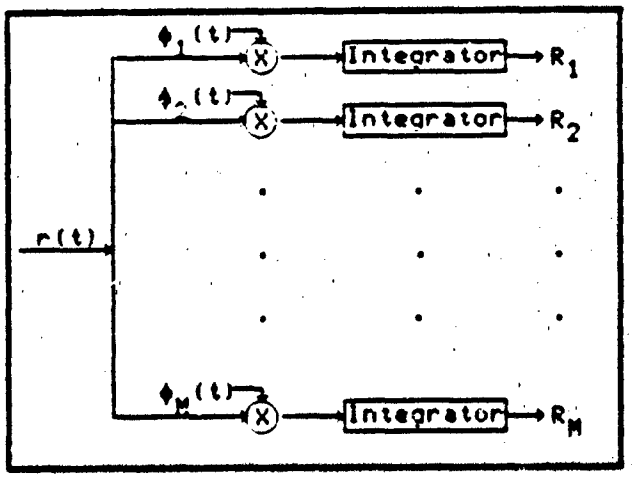

Figure 13. General Orthogonalization Implementation
Source: [6:228]

Figure 13 illustrates the implementation of the orthogonalization process. Each individual observation is developed by the basis function, $\phi_m(t)$, and the total dimension of the vector observation corresponds, at most, to the number of signals at the source. For the estimation problem at hand, the nature of the source is known and one-dimensional (singular valued). The random variable R_1 will be the only output and is a sufficient representation for the energy time function, $r_1(t)$, as all other R's are independent of R_1 (with white Gaussian noise assumed). Further, the basis function, $\phi_1(t)$ is [6:267]:

$$\phi_1(t) = s(t)/\text{sqrt}(E) \qquad (14)$$

where:

$$E = \int s^2(t)dt = \text{signal energy}$$

sqrt( ) = square root operation

integration is taken over a period

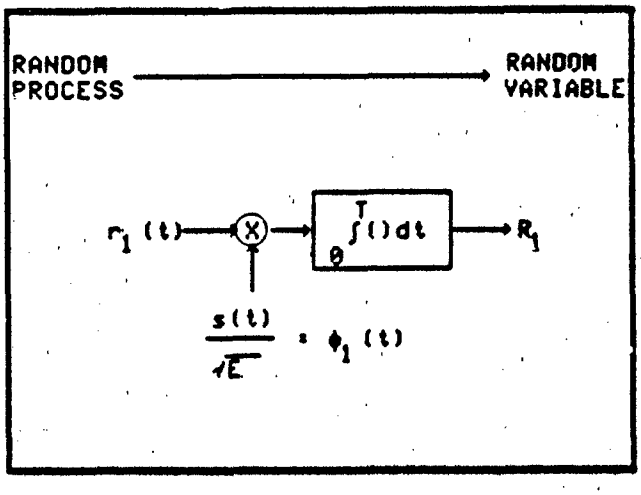

Figure 14. Monopulse Orthogonalization

Substitution of equations 11 and 14 into equation 13 yields:

$$R_1 = \int r_1(t)\phi_1(t)dt$$
$$= \int [AV_s s(t) + n_1(t)]\phi_1(t)dt$$
$$= \int [AV_s s^2(t)/\text{sqrt}(E)]dt + N_1$$
$$= AV_s \text{sqrt}(E) + N_1 \qquad (15)$$

where:

$$N_1 = \int s(t)n_1(t)/\text{sqrt}(E)dt$$

integrations are taken over a period

A similar development for R_2 yields:

$$R_2 = AV_d \text{sqrt}(E) + N_2 \qquad (16)$$

To form the joint density function required to characterize the received waveforms, the first and second moments of the vectors R_1 and R_2 must be determined. For the R_1 vector, and conditioning on the amplitude and angle of arrival (conditioning is performed on the angle of arrival rather than the received sum channel voltage because the sum voltage antenna pattern is a function of θ, therefore $E\{R_1/A,V_s\}$ is proportional to $E\{R_1/A,\theta\}$):

$$E\{R_1/A,\theta\} = AV_s \text{sqrt}(E) + E\{\int s(t)n_1(t)/\text{sqrt}(E)dt\}$$
$$= AV_s \text{sqrt}(E) + \int E\{s(t)\}E\{n_1(t)\}/\text{sqrt}(E)dt$$
$$= AV_s \text{sqrt}(E) \qquad (17)$$

where:

signal and noise are independent

mean of the noise is assumed zero

$E\{\ \}$ refers to the expectation operation

where: (continued)

integrations are taken over a period

A similar development for the R_2 vector yields:

$$E\{R_2/A,\theta\} = AV_d\,\mathrm{sqrt}(E) \tag{18}$$

where:

$E\{R_2/A,V_d\}$ is proportional to $E\{R_2/A,\theta\}$

$$\begin{aligned} \mathrm{var}\{R_1/A,\theta\} &= E\{[R_1 - AV_s\,\mathrm{sqrt}(E)]^2/A,\theta\} \\ &= E\{(R_1)^2/A,\theta\} - [AV_s\,\mathrm{sqrt}(E)]^2 \\ &= E\{(AV_s\,\mathrm{sqrt}(E) + N_1)^2/A,\theta\} - [AV_s\,\mathrm{sqrt}(E)]^2 \end{aligned} \tag{19}$$

Expanding the first term of equation 19:

$$E\{[AV_s\,\mathrm{sqrt}(E)]^2/A,\theta\} + E\{2AV_s\,\mathrm{sqrt}(E)N_1/A,\theta\} + E\{N_1^2/A,\theta\}$$

$$= [AV_s\,\mathrm{sqrt}(E)]^2 + E\{(\iint n_1(t)n_1(u)s(t)s(u)/E)\,dtdu/A,\theta\} \tag{19A}$$

Substituting equation 19A back into equation 19:

$$\begin{aligned} \mathrm{var}\{R_1/A,\theta\} &= 1/E \iint E\{n_1(t)n_1(u)\}E\{s(t)s(u)\}\,dtdu \\ &= N_0/2E \iint \delta(t-u)s(t)s(u)\,dtdu \\ &= N_0/2E \int s(u)s(u)\,du \\ &= N_0E/2E = N_0/2 \end{aligned} \tag{20}$$

where:

$\int \delta(t-u)dt = 1$; iff $t = u$ (sifting property)

$= 0$; otherwise

integrations are taken over a period

each noise sample is independent, Gaussian, zero mean

autocorrelation of the noise = $(N_0/2)\delta(\tau)$

A similar development for R_2 yields:

$$\text{var}\{R_2/A,\theta\} = N_0/2 \tag{21}$$

$$\text{cov}\{R_1,R_2/A,\theta\} = E\{(R_1R_2)^2)/A,\theta\} - A^2EV_sV_d \tag{22}$$

Expanding the first term of equation 22:

$$E\{[AV_s\,\text{sqrt}(E) + N_1][AV_d\,\text{sqrt}(E) + N_2]/A,\theta\}$$
$$= A^2V_sV_dE + AV_s\,\text{sqrt}(E)E\{N_2/A,\theta\} + AV_d\,\text{sqrt}(E)E\{N_2/A,\theta\} + E\{N_1N_2/A,\theta\}$$
$$= A^2V_sV_dE \tag{22A}$$

Substituting equation 22A for the first term of equation 22:

$$\text{cov}\{R_1,R_2/A,\theta\} = 0 \tag{23}$$

Equation 23 shows the vector observations to be uncorrelated. Further, with Gaussian noise assumed and linear operations to develop the vector observations, R_1 and R_2 will be Gaussian and uncorrelated, therefore independent [7:199]. The receiver observation is then the joint density of two independent Gaussian random variables. Independence implies that the joint density of R_1 and R_2 is the product of the marginal R_1 and R_2 distributions. With Gaussian distributions and statistics given by equations 17, 18, 20, 21, and 23 the joint conditional density of the receiver observation is:

$$p_R(r/A,\theta) = p(r_1/A,\theta)p(r_2/A,\theta)$$
$$= [piN_0]^{-1}\exp(-1/N_0[(r_1 - AV_s\,\text{sqrt}(E))^2 + (r_2 - AV_d\,\text{sqrt}(E))^2]) \tag{24}$$

where:

$$R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}, \quad r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

Maximum-Likelihood (ml) estimation

The maximum-likelihood approach is to choose the estimate for the unknown parameter to be the parameter value that most likely caused a given observation to occur. In the general case, the joint conditional density of the receiver observation is denoted as the likelihood function [5:65]. Another useful function for monotonic functions (1 to 1 mapping between the function and the likelihood function) is the logarithm of the likelihood function. With a 1 to 1 mapping, working with the log-likelihood function is equivalent to working with the likelihood function [8:181]. The maximum-likelihood (ml) estimate is then that value of the unknown parameter for which the likelihood function is a maximum. Or equivalently, a necessary condition for the ml estimate is obtained by differentiating the log-likelihood function with respect to the unknown parameter and setting the result equal to zero [5:65].

Calling this condition the log-likelihood equation, it can be expressed as:

$$A'[\ln[p_R(r/A,\theta)]]\Big|_{A=\hat{A}_{ml}} = 0 \qquad (25)$$

where:

A'[] represents the partial wrt the parameter A

Maximum-likelihood estimates can then be obtained from unique solutions of equation 25. To apply the procedure described by equation 25 to the conditional density of equation 24, the log-likelihood equation must first be established. Using equation 24:

$$\ln[p_R(r/A,\theta)] = -1/N_0[(r_1 - AV_s\,\mathrm{sqrt}(E))^2 + (r_2 - AV_d\,\mathrm{sqrt}(E))^2] - \ln[(pi)N_0]$$

$$= -1/N_0[(r_1)^2 - 2Ar_1V_s\,\mathrm{sqrt}(E) + (AV_s\,\mathrm{sqrt}(E))^2]$$

$$-1/N_0[(r_2)^2 - 2Ar_2V_d\,\mathrm{sqrt}(E) + (AV_d\,\mathrm{sqrt}(E))^2]$$

$$-\ln[(pi)N_0]$$

Taking the partial wrt the parameter A and setting the result equal to zero will provide the log-likelihood equation described by equation 25:

$$0 = 2r_1V_s\,\mathrm{sqrt}(E)/N_0 - 2A(V_s\,\mathrm{sqrt}(E))^2/N_0 +$$

$$2r_2V_d\,\mathrm{sqrt}(E)/N_0 - 2A(V_d\,\mathrm{sqrt}(E))^2/N_0 \qquad (25A)$$

$$= r_1V_s\,\mathrm{sqrt}(E) - A(V_s)^2E + r_2V_d\,\mathrm{sqrt}(E) - A(V_d)^2E$$

Solving for $\hat{A}_{ml}$:

$$\hat{A}_{ml} = \frac{r_1V_s\,\mathrm{sqrt}(E) + r_2V_d\,\mathrm{sqrt}(E)}{(V_s)^2E + (V_d)^2E}$$

$$= \frac{1}{\mathrm{sqrt}(E)}\left[\frac{r_1V_s + r_2V_d}{(V_s)^2 + (V_d)^2}\right] \qquad (26)$$

The other unknown parameter of interest is the angle of arrival. An estimate of θ allows boresight movement to keep the tracking axis (boresight) upon the target. The log-likelihood function terms that are functions of the angle θ are V_s and V_d which are described by equations 9 and 10, respectively. Taking the partial with respect to the parameter θ and setting the result equal to zero yields:

$$0 = 2Ar_1V_s'\,\mathrm{sqrt}(E)/N_0 - 2AV_s\,\mathrm{sqrt}(E)AV_s'\,\mathrm{sqrt}(E)/N_0 +$$

$$2Ar_2V_d'\,\mathrm{sqrt}(E)/N_0 - 2AV_d\,\mathrm{sqrt}(E)AV_d'\,\mathrm{sqrt}(E)/N_0 \qquad (27)$$

$$= r_1V_s' - AV_sV_s'\,\mathrm{sqrt}(E) + r_2V_d' - AV_dV_d'\,\mathrm{sqrt}(E) \qquad (27A)$$

where

V_s', V_d' represents the partial wrt the parameter θ

Substituting equation 26 for A of equation 27A and simplifying:

$$0 = r_1 V_s' V_d^2 - r_2 V_s V_d V_s' + r_2 V_d' V_s^2 - r_1 V_s V_d V_d'$$

or equivalently:

$$r_1 V_d [V_s' V_d - V_s V_d'] = r_2 V_s [V_s' V_d - V_s V_d']$$

equating coefficients, and evaluating at the position estimate:

$$r_1 V_d(\hat{\theta}) - r_2 V_s(\hat{\theta}) = 0 \qquad (28)$$

The maximum-likelihood estimation procedure has generated two estimates. The first estimate, equation 26, describes the most likely amplitude for a given observation. The second estimate, equation 28, was derived from the amplitude estimate and gives a relationship that contains both observation terms and amplitude weighted angle terms. In order to get a better feel for the relationship between equation 28 and the estimate for target position, it is necessary to look more closely at equation 28.

For the second estimate (equation 28) the random quantities will be the noise corrupted observations, R_1 and R_2. In a no-noise environment, the observations R_1 and R_2 would equal their respective mean values and equation 28 restated for the no-noise case is:

$$0 = [AV_s(\theta)\text{sqrt}(E)]V_d(\hat{\theta}) - [AV_d(\theta)\text{sqrt}(E)]V_s(\hat{\theta}) \qquad (29)$$

where:

$V_s(\hat{\theta})$, $V_d(\hat{\theta})$ are functions of an estimated target position, $\hat{\theta}$

$V_s(\theta)$, $V_d(\theta)$ are functions of the true target position, θ

Substitution of equations 9 and 10 into equation 29 yields:

$$0 = A\sqrt{2E}/2[(1 + \cos(2.1\theta/BW))\sin(1.88\hat{\theta}/BW) - (1 + \cos(2.1\hat{\theta}/BW))\sin(1.88\theta/BW)] \quad (29A)$$

Inspection of equation 29A indicates that the equality will be satisfied, in the no-noise case, when the estimate for theta equals the parameter value. For all other estimated values of θ, unequal to the true target position, the equation will not be zero but will be an error that is proportional to the difference of the estimate from the actual value. Further, with no-noise assumed, the target position could be determined exactly with one observed pulse (ie the required estimate for target position to force equation 29A to zero) and the boresight could then move to the exact target location. With additive noise, equation 28 will not be equal to zero and the maximum-likelihood estimate for target position will be in error. In this light, the second estimate (equation 28), is the error equation for the amplitude-comparison monopulse radar. The derived second estimate will be used to move the boresight to the vicinity of the target.

In order to bound the estimation error for the target position, it is first necessary to compute the statistics for the generated estimate for the amplitude of the return signal.

$$E\{\hat{A}_{ml}/A,\theta\} = \frac{V_s E\{R_1\} + V_d E\{R_2\}}{\text{sqrt}(E)[(V_s)^2 + (V_d)^2]}$$

$$= \frac{V_s(AV_s\,\text{sqrt}(E)) + V_d(AV_d\,\text{sqrt}(E))}{\text{sqrt}(E)[(V_s)^2 + (V_d)^2]}$$

$$= A \qquad (30)$$

Equation 30 indicates that for all values of A, the average value of the estimate equals the parameter that we are trying to estimate. With this relationship satisfied, the estimate is commonly referred to as unbiased [5:64].

$$\text{var}\{\hat{A}_{ml}/A,\theta\} = E\{(\hat{A}_{ml})^2\} - E\{(\hat{A}_{ml})\}^2$$

$$= E\left\{\frac{1}{E\left[\frac{R_1V_s + R_2V_d}{(V_s)^2 + (V_d)^2}\right]^2}\right\} - A^2$$

Let $k = 1/(E[(V_s)^2 + (V_d)^2])$

$$\text{var}\{\hat{A}_{ml}/A,\theta\} = kE\{(R_1V_s)^2 + 2R_1R_2V_sV_d + (R_2V_d)^2\} - A^2$$

$$= k[V_s^2(\text{var}(R_1)+A^2V_s^2E) + 2A^2EV_s^2V_d^2 + V_d^2(\text{var}(R_2)+A^2V_d^2E)] - A^2$$

$$= k[V_s^2N_0/2 + V_d^2N_0/2 + (A^2V_s^4E + 2A^2V_s^2V_d^2E + A^2V_d^4E)] - A^2$$

$$= kN_0/2[V_s^2 + V_d^2] + A^2kE[V_s^2 + V_d^2]^2 - A^2$$

$$\text{var}\{\hat{A}_{ml}/A,\theta\} = \frac{N_0}{2E(V_s^2 + V_d^2)} + A^2 - A^2$$

$$= \frac{N_0}{2E(V_s^2 + V_d^2)} \qquad (31)$$

Cramer-Rao Lower Bound (CRB)

For any unbiased estimate $\hat{A}$ of a scalar A, the conditional variance is bounded by [8:232]:

$$\text{var}(\hat{A}/A) \geq [E\{(A'[\ln(p_R(r/A))])^2\}]^{-1}$$

or equivalently, $$\text{var}(\hat{A}/A) \geq [-E\{A''[\ln(p_R(r/A))]\}]^{-1} \quad (32)$$

where:

A' represents the partial wrt the parameter A

A" represents the second partial wrt the parameter A

Equation 32 is usually referred to as the Cramer-Rac inequality. Consider applying the steps described by equation 32 to the amplitude estimate:

$$A'[\ln(p_R(r/A))] = 2r_1V_s\text{sqrt}(E)/N_0 - 2AV_s^2E/N_0 + 2r_2V_d\text{sqrt}(E)/N_0 - 2AV_d^2E/N_0 \quad (33)$$

$$-A''[\ln(p_R(r/A))] = 2V_s^2E/N_0 + 2V_d^2E/N_0 \quad (33A)$$

$$E\{-A''[\ln(p_R(r/A))]\}^{-1} = \frac{N_0}{2E(V_s^2 + V_d^2)}$$

Using the inequality given by equation 32:

$$\text{var}(\hat{A}_{ml}/A,\theta) \geq \frac{N_0}{2E(V_s^2 + V_d^2)} \quad (34)$$

Equation 31 previously defined the conditional variance of the amplitude estimate. Comparison of equation 31 with the conditional inequality of equation 34 indicates that the conditional variance satisfies the CRB with equality. Any unbiased estimate that satisfies

the inequality of equation 32 with an equality is called an efficient estimate [5:66]. There is no guarantee that an efficient estimator exists for a given problem, however, if one does exist, the maximum-likelihood estimate will be efficient [8:233]. Efficiency is of importance because equality with the lower bound described by equation 32 assures minimization of the conditional variance of the estimate, and therefore, provides the best estimate available. Applying equation 32 to an estimate of target location, $\hat{\theta}$:

$$\theta'[\ln p_R(r/A,\theta)] = 2Ar_1V_s'\,\mathrm{sqrt}(E)/N_0 - 2AV_s\,\mathrm{sqrt}(E)AV_s'\,\mathrm{sqrt}(E)/N_0 + 2Ar_2V_d'\,\mathrm{sqrt}(E)/N_0 - 2AV_d\,\mathrm{sqrt}(E)AV_d'\,\mathrm{sqrt}(E)/N_0$$

$$\theta''[\ln p_R(r/A,\theta)] = 2Ar_1V_s''\,\mathrm{sqrt}(E)/N_0 - 2A^2(V_s')^2E/N_0 - 2A^2V_sV_s''E/N_0 + 2Ar_2V_d''\,\mathrm{sqrt}(E)/N_0 - 2A^2(V_d')^2E/N_0 - 2A^2V_dV_d''E/N_0$$

$$= 2A\,\mathrm{sqrt}(E)/N_0[V_s''(r_1 - AV_s\,\mathrm{sqrt}(E)) + V_d''(r_2 - AV_d\,\mathrm{sqrt}(E))] - 2A^2E/N_0[(V_s')^2 + (V_d')^2]$$

$$= 2A\,\mathrm{sqrt}(E)/N_0[V_s''N_1 + V_d''N_2 - A\,\mathrm{sqrt}(E)(V_s')^2 - A\,\mathrm{sqrt}(E)(V_d')^2] \quad (35)$$

$$-E\{\theta''[\ln p_R(r/A,\theta)]\} = 2A^2E/N_0[(V_s')^2 + (V_d')^2] \quad (35A)$$

where:

$$E\{V_s''N_1\} = E\{V_d''N_2\} = 0$$

$$[-E\{\theta''[\ln p_R(r/A,\theta)]\}]^{-1} = \frac{N_0}{2A^2E}\left[\frac{1}{(V_s')^2 + (V_d')^2}\right] \quad (35B)$$

Substitution of equations 9 and 10 into equation 35B and performing the indicated first partial squared, and then simplifying yields the final form for the Cramer-Rao bound (CRB) for the conditional variance of the unbiased estimate $\hat{\theta}$:

$$\text{var}(\hat{\theta}/A,\theta) \geq \frac{N_0[BW]^2}{2A^2E}\left[\frac{1}{2.87 + 1.8\cos(3.8\theta/BW) - 1.1\cos(4.2\theta/BW)}\right] \quad (36)$$

Equation 36 is one of the bounds for the tracking error of the monopulse radar. Application of the conditional variance as a mean square error implies a priori knowledge about the average error. Specifically, a zero average error is implied. This will clearly not be the case if the estimate is no longer unbiased, and at that point this form of the CRB loses usefulness. Before leaving the CRB, a few comments about equation 36 are in order. Inspection of the leading coefficient indicates that the bound is directly proportional to the beamwidth and inversely proportional to the signal-to-noise ratio. This result is intuitively appealing, as the bound predicts a small tracking error in a low-noise (high SNR, low BW) environment. Note also that if the converse is true (low SNR, high BW), then the variance of the estimated position grows quite large. For low SNR, the CRB does not adequately approximate the estimation error because the likelihood function can have several peaks and maximization of the likelihood function may only produce a local, instead of absolute, maximum. Errors are then made on the likelihood function sidelobe peaks [9:148].

Ziv-Zakai Lower Bound (ZZB)

The Ziv-Zakai bound is derived by comparing the estimation problem with the optimal detection problem. The resulting bounds are independent of bias and explicitly include the dependence on the a priori interval of the desired parameter [10:386]. The estimation decision rule will have an associated probability of error (P_e) that will be lower bounded by the P_e associated with an optimal detection scheme. This lower bound (estimation decision vs optimal decision) enables the derivation of performance bounds that are based upon detection theory.

Consider an estimation technique for the target angle off beam-axis, θ_t, of a received message when it is known that the angle is either θ_0 or θ_1. The Ziv-Zakai approach is to compare the angle estimate, $\hat{\theta}$, with an average value (ie compare $\hat{\theta}$ with $(\theta_0 + \theta_1)/2$) and then decide between θ_0 and θ_1.

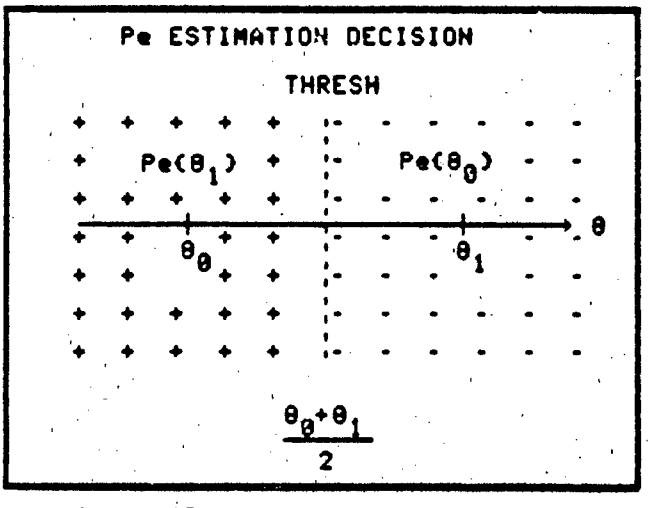

Figure 15. Estimation Decision Space
Source: [10:387]

From the law of total probability:

$$P_e(est) = P\{\theta_0\}P\{(\hat{\theta} > \frac{\theta_0 + \theta_1}{2})/\theta_0\} + P\{\theta_1\}P\{(\hat{\theta} < \frac{\theta_0 + \theta_1}{2})/\theta_1\}$$

$$= 1/2P\{(\hat{\theta} > \theta_0/2 + \theta_1/2)/\theta_0\} + 1/2P\{(\hat{\theta} < \theta_0/2 + \theta_1/2)/\theta_1\}$$

where:

$P\{\theta_0\} = P\{\theta_1\}$

conditioning on θ_0 assumes $s(\theta_0)$ was transmitted

Let: $d = \theta_1 - \theta_0$

$$P_e(est) = 1/2P\{(\hat{\theta} > \theta_0 + d/2)/\theta_0\} + 1/2P\{(\hat{\theta} < \theta_1 - d/2)/\theta_1\}$$

$$= 1/2P\{(\hat{\theta} - \theta_0 > d/2)/\theta_0\} + 1/2P\{(\hat{\theta} - \theta_1 < -d/2)/\theta_1\}$$

$$= 1/2P\{(|\hat{\theta} - \theta_0| \geq d/2)/\theta_0\} + 1/2P\{(|\theta_1 - \hat{\theta}| \geq d/2)/\theta_1\}$$

Recall that P_e must be greater than an optimal detection scheme. (ie for equiprobable binary signaling):

$$P_e \leq 1/2P\{(|\hat{\theta} - \theta_0| \geq d/2)/\theta_0\} + 1/2P\{(|\hat{\theta} - \theta_1| \geq d/2)/\theta_1\} \quad (37)$$

where

$$|\hat{\theta} - \theta_1| = |\theta_1 - \hat{\theta}|$$

To proceed, the right side of equation 37 must be simplified. Reference 9 applied the Tchebycheff inequality to the right side of equation 37. The Tchebycheff inequality is a specific case of a more general rule which is termed the inequality of Bienamye' [11:151]:

Let $Y = |x - a|^n$; since Y is always positive:

$$P\{|x-a| \geq b\} \leq E\{|x - a|^n\}/b^n \quad (38)$$

Applying equation 38 to equation 37:

Let: $x = \hat{\theta}$; $d/2 = b$; $a = \theta_0$ or θ_1; $n = 2$

$$P\{|\hat{\theta} - \theta_0| \geq d/2\} \leq E\{|\hat{\theta} - \theta_0|^2\}/(d/2)^2 \quad (38A)$$

$$P\{|\hat{\theta} - \theta_1| \geq d/2\} \leq E\{|\hat{\theta} - \theta_1|^2\}/(d/2)^2 \quad (38B)$$

Using equations 38A and 38B in the inequality of equation 37:

$$P_e \leq 1/2E\{|\hat{\theta} - \theta_0|^2/\theta_0\}/(d/2)^2 + 1/2E\{|\hat{\theta} - \theta_1|^2/\theta_1\}/(d/2)^2$$

$$[d^2/4]P_e \leq 1/2[E\{(\hat{\theta} - \theta_0)^2/\theta_0\} + E\{(\hat{\theta} - \theta_1)^2/\theta_1\}]$$

$$[d^2/4]P_e \leq 1/2[\hat{e}^2(\theta_0) + \hat{e}^2(\theta_1)] \quad (39)$$

where:

$E\{(\hat{\theta} - \theta)^2/\theta\} = \hat{e}^2(\theta)$ = mean-square estimation error

To simplify, assume $\theta_0 = -\theta_1$:

$$[(2\theta)^2/4]P_e(\theta,-\theta) \leq 1/2[E\{(\hat{\theta} - \theta)^2/\theta\} + E\{(\hat{\theta} - \theta)^2/-\theta\}] \quad (40)$$

where:

$P_e(\theta,-\theta)$ = error probability of the best procedure for deciding whether a target is at θ, or $-\theta$, when it is known to be at one of these positions with equal probability

$$\theta^2 P_e(\theta,-\theta) \leq 1/2\hat{e}^2(\theta) + 1/2\hat{e}^2(-\theta) \quad (40A)$$

The right hand side of the inequality of equation 40A is a lower bound to the arithmetic average of the mean square estimation error ($\hat{e}^2$) for any pair of the values of the parameter θ_t which are 2θ units apart [10:387]. Imposing the symmetry of the mean-square errors equation 40A then becomes:

$$\theta^2 P_e(\theta,-\theta) \leq 1/2\hat{e}^2(\theta) + 1/2\hat{e}^2(\theta)$$

$$\hat{e}^2 \geq \theta^2 P_e(\theta,-\theta)$$

Many bounds on the estimation error can be derived from this basic result. Clearly, the worst case error will be lower bounded by the maximum in the a priori interval of the parameter θ_t [12:650]. In this case, the estimation error is bounded from below by the scaled maximum of the product of the square of the difference of the two possible angles and the minimum achievable detection error for equiprobable binary signaling over the a priori interval.

$$\hat{e}^2 \geq \max_{0 < \theta_t < \theta_{max}} \theta_t^2 P_e(\theta_t,-\theta_t) \qquad (40B)$$

Reference 9 developed an extension to the bound on the mean-square error for bearing estimation given in equation 40B for the specific case of maximum-likelihood estimation. The development is as follows [9:156]:

> A tighter bound can be obtained for the case where the probability density function is a function only of the difference between estimated and actual parameter values. This constraint is satisfied by a maximum-likelihood estimator $u = \sin(\theta)$. Realizing that $|\sin(\theta_0) - \sin(\theta_1)| \leq |\theta_0 - \theta_1|$, the maximum-likelihood estimator can be applied to give the appropriate bounds on the mean-square error:

$$\hat{e}^2 \geq \max_{0 \leq \theta_t \leq \theta_p} [\sin(\theta_t)]^2 P_e(\theta_t, -\theta_t) \qquad (40C)$$

where:

$\sin(\theta_p) = [\sin(\theta_{max}) + \sin(\theta)]/2$

θ_{max} = maximum possible angle (a priori interval)

Equation 40C, [9:156], will be the form of the Ziv-Zakai bound used to predict the mean-square estimation error for a maximum-likelihood estimate of azimuth angle. To apply equation 40C, the error probabiltiy, P_e, must first be computed.

H_0 and H_1 are the two hypotheses for the binary detection problem of whether the azimuth angle of the target is at θ_0 or θ_1, when it is known that a target is at one of these two positions. The two hypotheses are:

H_0: $R = E\{R/H_0\} + N$

H_1: $R = E\{R/H_1\} + N$

The total probability of error is:

P_e = P{making an incorrect decision}

= P{deciding H_0 but H_1 is true or deciding H_1 but H_0 is true}

$$= P(H_1) \int_{H1} p_R(r/H_0)dr + P(H_0) \int_{H0} p_R(r/H_1)dr$$

where:

H_1 integral is area under "0" density in the "1" region

H_0 integral is area under "1" density in the "0" region

The decision rule is then:

$$L(R) \underset{H0}{\overset{H1}{\gtrless}} \frac{P(H_0)}{P(H_1)} = 1$$

where:

L(R) is the likelihood ratio [5:26] $= \dfrac{p(r/H_1)}{p(r/H_0)}$

equal a priori probabilities is assumed

equivalently:

$$l(R) \underset{H0}{\overset{H1}{\gtrless}} 0$$

where:

l(R) is the natural logarithm of the likelihood ratio

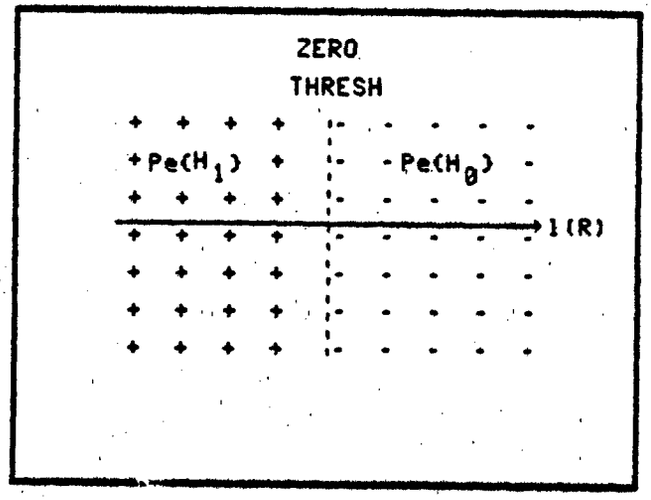

Figure 16. Decision Space for l(R)

From Figure 16, the total probability of error is:

$$P_e = 1/2P\{l(R) > 0/H_0\} + 1/2P\{l(R) < 0/H_1\} \quad (41)$$

The log-likelihood ratio is a random variable. This is easily seen from the derivation for the likelihood ratio which involves the ratio of two functions of a random variable; therefore, L(R) is a one-dimensional random variable [5:26], and the logarithm of a random variable is random. With Gaussian statistics for each of the joint conditional densities, l(R) is the sum of independent Gaussian random variables; therefore Gaussian. Computing the statistics of the one-dimensional Gaussian random variable l(R) will allow the development of the probability of error as described by equation 41. From equation 24, the joint conditional density when H_0 is true is:

$$p_R(r/H_0) = \text{equation 24 evaluated at } \theta = \theta_0 \quad (42)$$

$$= \frac{(1)\exp(-1/N_0[(r_1-AV_s(\theta_0)\text{sqrt}(E))^2 + (r_2-AV_d(\theta_0)\text{sqrt}(E))^2])}{(\text{pi})N_0} \quad (42A)$$

Similarly, the joint conditional density when H_1 is true is:

$$= \frac{(1)\exp(-1/N_0[(r_1-AV_s(\theta_1)\text{sqrt}(E))^2 + (r_2-AV_d(\theta_1)\text{sqrt}(E))^2])}{(\text{pi})N_0} \quad (42B)$$

Then l(R) is the logarithm of equation 42B divided by equation 42A:

$$l(R) = -1/N_0(r_1-a)^2 - 1/N_0(r_2-b)^2 + 1/N_0(r_1-c)^2 + 1/N_0(r_2-d)^2 \quad (43)$$

where:

$$a = E\{R_1/H_1\} = AV_s(\theta_1)\text{sqrt}(E)$$

$$b = E\{R_2/H_1\} = AV_d(\theta_1)\text{sqrt}(E)$$

$$c = E\{R_1/H_0\} = AV_s(\theta_0)\text{sqrt}(E)$$

$$d = E\{R_2/H_0\} = AV_d(\theta_0)\text{sqrt}(E)$$

$$l(R) = -1/N_0(r_1^2 - 2r_1a + a^2) - 1/N_0(r_2^2 - 2r_2b + b^2)$$
$$+1/N_0(r_1^2 - 2r_1c + c^2) + 1/N_0(r_2^2 - 2r_2d + d^2)$$
$$= 1/N_0[2r_1(AV_s(\theta_1)\text{sqrt}(E) - AV_s(\theta_0)\text{sqrt}(E)) +$$
$$2r_2(AV_d(\theta_1)\text{sqrt}(E) - AV_d(\theta_0)\text{sqrt}(E)) -$$
$$A^2EV_s^2(\theta_1) + A^2EV_s^2(\theta_0) - A^2EV_d^2(\theta_1) + A^2EV_d^2(\theta_0)]$$
$$= 2A\text{sqrt}(E)/N_0[r_1(V_s(\theta_1) - V_s(\theta_0)) + r_2(V_d(\theta_1) - V_d(\theta_0)) +$$
$$A\text{sqrt}(E)/2(V_s^2(\theta_0) - V_s^2(\theta_1) + V_d^2(\theta_0) - V_d^2(\theta_1)] \quad (43A)$$

Introducing a new version of $l(R)$:

$$l'(R) = l(R)/(2A\text{sqrt}(E)/N_0) \quad (43B)$$

The statistics of $l'(R)$ for the hypothesis H_0 are:

$$E\{l'(R)/H_0\} = E\{R_1(V_s(\theta_1) - V_s(\theta_0)/H_0)\} + E\{R_2(V_d(\theta_1) - V_d(\theta_0)/H_0)\} +$$
$$A\text{sqrt}(E)/2(V_s^2(\theta_0) - V_s^2(\theta_1) + V_d^2(\theta_0) - V_d^2(\theta_1))$$
$$= A\text{sqrt}(E)[V_s(\theta_0)(V_s(\theta_1) - V_s(\theta_0)) + V_d(\theta_0)(V_d(\theta_1) - V_d(\theta_0))] +$$
$$A\text{sqrt}(E)/2(V_s^2(\theta_0) - V_s^2(\theta_1) + V_d^2(\theta_0) - V_d^2(\theta_1))$$
$$= A\text{sqrt}(E)[V_s(\theta_0)V_s(\theta_1) + V_d(\theta_0)V_d(\theta_1) -$$
$$A\text{sqrt}(E)/2[(V_s^2(\theta_0) + V_s^2(\theta_1) + V_d^2(\theta_0) + V_d^2(\theta_1)]$$
$$= A\text{sqrt}(E)W/2[p - 1.0] \quad (44)$$

where:

$$W = V_s^2(\theta_0) + V_s^2(\theta_1) + V_d^2(\theta_0) + V_d^2(\theta_1) \quad (45)$$

$$p = 2/W[V_s(\theta_0)V_s(\theta_1) + V_d(\theta_0)V_d(\theta_1)] \quad (46)$$

$$\text{var}\{l'(R)/H_0\} = \text{var}\{R_1/H_0\}[V_s(\theta_1) - V_s(\theta_0)]^2 +$$
$$\text{var}\{R_2/H_0\}[V_d(\theta_1) - V_d(\theta_0)]^2$$
$$= N_0/2[V_s^2(\theta_0) + V_s^2(\theta_1) + V_d^2(\theta_0) + V_d^2(\theta_1)] -$$
$$N_0[V_s(\theta_0)V_s(\theta_1) + V_d(\theta_0)V_d(\theta_1)]$$

$$\text{var}\{l'(R)/H_0\} = N_0W/2[1.0 - p] \quad (47)$$

A similar development for hypothesis H_1 yields:

$$E\{l'(R)/H_1\} = A\text{sqrt}(E)W/2[1.0 - p] \quad (48)$$

$$\text{var}\{l'(R)/H_1\} = N_0W/2[1.0 - p] \quad (49)$$

The conditional distribution under the hypothesis H_1 is:

$$p(l'(R)/H_1) = k\exp[-(l'(R)-E\{l'(R)/H_1\})^2/2\text{var}\{l'(R)/H_1\}]$$

where:

$$k = [\text{sqrt}(2(\text{pi})\text{var}\{l'(R)/H_1\})]^{-1}$$

$$E\{l'(R)/H_1\} = A\text{sqrt}(E)W/2[1.0 - p]$$

$$\text{var}\{l'(R)/H_1\} = N_0W/2[1.0 - p]$$

Similarly, under the hypothesis H_0:

$$p(l'(R)/H_0) = k\exp[-(l'(R)+E\{l'(R)/H_1\})^2/2\text{var}\{l'(R)/H_1\}]$$

where:

$$E\{l'(R)/H_0\} = -E\{l'(R)/H_1\}$$

$$\text{var}\{l'(R)/H_0\} = \text{var}\{l'(R)/H_1\}$$

Equation 41 then becomes:

$$P_e = 1/2P\{l'(R) > 0/H_0\} + 1/2P\{l'(R) < 0/H_1\}$$

$$= 1/2P\{l'(R) > 0/H_0\} + 1/2P\{l'(R) > 0/H_0\}$$

$$= P\{l'(R) > 0/H_0\}$$

where:

Due to the symmetry of the shifted distributions:

$$P\{l'(R) > 0/H_0\} = P\{l'(R) < 0/H_1\}$$

$$P_e = \int p(l'(R)/H_0)dl'(R)$$

$$= \int k\exp[-(l'(R)+E\{l'(R)/H_1\})^2/2\mathrm{var}\{l'(R)/H_1\}]dl'(R) \qquad (50)$$

where

integrations range from zero to infinity

Let $u = (l'(R)+E\{l'(R)/H_1\})/\mathrm{sqrt}(\mathrm{var}\{l'(R)/H_1\})$

$du = dl'(R)/\mathrm{sqrt}(\mathrm{var}\{l'(R)/H_1\})$; $dl'(R) = \mathrm{sqrt}(\mathrm{var}\{l'(R)/H_1\})du$

when $l'(R) = 0$; $u = E\{l'(R)/H_1\}/\mathrm{sqrt}(\mathrm{var}\{l'(R)/H_1\})$

Equation 50 then becomes:

$$P_e = \int_{l'(R)=0}^{\infty} \mathrm{sqrt}(2(pi))^{-1}\exp[-u^2/2]du$$

$$= Q[u \text{ when } l'(R) = 0]$$

where:

$$Q[a] = \mathrm{sqrt}(2(pi))^{-1}\int_a^{\infty} \exp[-x^2/2]dx \qquad [8:257]$$

$$P_e = Q[E\{l'(R)/H_1\}/\mathrm{sqrt}(\mathrm{var}\{l'(R)/H_1\})]$$

$$= Q[A\mathrm{sqrt}(E)W/2[1.0 - p]/\mathrm{sqrt}(N_0W/2[1.0 - p])$$

$$= Q[A\mathrm{sqrt}([EW/2N_0](1.0 - p))] \qquad (51)$$

Equation 51 describes the necessary P_e of the best procedure for deciding whether a target is a θ_0 or θ_1, when it is known to be at one of these two positions with equal probability. For the case at hand, $\theta_0 = -\theta_1$ and substitution of equations 9 and 10 into equations 45 and 46 the factors W and p become:

$$W = (V_s^2(\theta_0) + V_s^2(\theta_1) + V_d^2(\theta_0) + V_d^2(\theta_1)$$

$$= 1/2[1+2\cos(2.1\theta/BW) + \cos^2(2.1\theta/BW)] + \sin^2(1.9\theta/BW) +$$

$$1/2[1+2\cos(-2.1\theta/BW) + \cos^2(-2.1\theta/BW)] + \sin^2(-1.9\theta/BW)$$

$$= 5/2 + 2\cos(2.1\theta/BW) + 1/2\cos(4.2\theta/BW) - \cos(3.8\theta/BW) \quad (52)$$

Figure 17 provides a plot of the factor W out to three beamwidths.

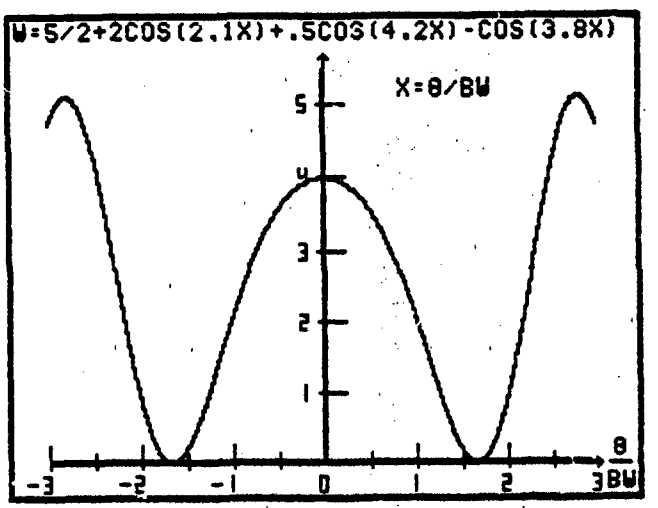

Figure 17. W vs Normalized Beamwidth

$$p = 2/W[V_s(\theta_0)V_s(\theta_1) + V_d(\theta_0)V_d(\theta_1)]$$

$$= [1/2 + 2\cos(2.1\theta/BW) + 1/2\cos(4.2\theta/BW) + \cos(3.8\theta/BW)]/W \quad (53)$$

A plot of the parameter p is provided in Figure 18. Inspection of Figure 18 shows the parameter p to be periodic in normalized beamwidth and approximately equal to cos(2.22θ/BW). Figure 18 displays only a small window of the function p because the particular form of equation 52 has discontinuities at multiples of 1.7 times the normalized beamwidth. No approximation for p is utilized because all angles of

interest are within plus or minus one-half the beamwidth. With the magnitude of p always less than or equal to 1, p satisfies the condition of a correlation coefficient.

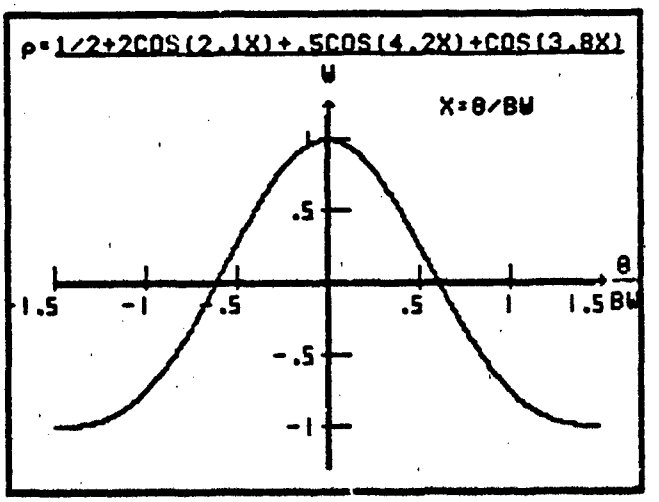

Figure 18. p vs Normalized Beamwidth

Inspection of Figures 17 and 18 reveal that when the parameter θ is zero, W equals 4 and p is unity. When p is unity, equation 51 is maximum; and therefore P_e is maximum. This result is intuitively appealing as targets with small seperation angles will be difficult to distinguish. Substitution of equation 51 into equation 40C, along with equations 52 and 53, will provide the Ziv-Zakai lower bound for the estimation error of a maximum-likelihood estimate of the target azimuth angle.

Tracking Loop

The output of the monopulse estimator was previously shown (equation 28) to be an error signal that is proportional to the difference of a noise corrupted estimate of target position and actual target location. Restating equation 28:

$$e = R_1(\theta)V_d(\hat{\theta}) - R_2(\theta)V_s(\hat{\theta}) \qquad (28)$$

To provide automatic tracking of targets, the error signal actuates a servo-control system to steer the beam-axis on target. If the error signal does not change rapidly with respect to the operating frequency of the radar, the error will be approximately a dc, or step, input to the antenna servo system. The time domain response of the servo is then approximately the response that would be obtained due to a scaled step function proportional to the error signal. In that light, the development for the tracking loop will be as follows:

1) A relationship will be derived between the error signal and a step function that is proportional to the difference of the estimated target location and actual target position. This conversion will be in the form of a discriminator curve that will, within limits, provide a 1 to 1 mapping between error and a step function with amplitude proportional to required boresight movement.
2) Since the error signal will be in the form of a step function, an improved Type I servo system will be developed to provide the desired response time and steady state error for the

monopulse tracker. A Type I servo gives zero steady state error for a step input, and an improved Type I provides the desired response time.

3) Relationships will be established for tracking moving targets.

Discriminator

With additive noise, the error equation is a random quantity as R_1 and R_2 are independent, Gaussian, distributed random variables. As stated earlier, the discriminator curve is established assuming a perfect mapping from error signal to a step function required to align the beam-axis with the target position (ie the mapping is accomplished in the no-noise environment). In this case, the observations are given by the conditional means:

$$R = E\{R(\theta)/A,\theta\}$$

$$= AV_s(\theta)\text{sqrt}(E) \text{ for observation } R_1 \quad (54)$$

$$= AV_d(\theta)\text{sqrt}(E) \text{ for observation } R_2 \quad (55)$$

Substitution of equations 54 and 55 into equation 28 yields:

$$e = AV_s(\theta)\text{sqrt}(E)V_d(\hat{\theta}) - AV_d(\theta)\text{sqrt}(E)V_s(\hat{\theta}) \quad (56)$$

Recall that $V_s(\theta)$ and $V_d(\theta)$ were described by equations 9 and 10:

$$V_s(\theta) = \text{sqrt}(2)/2[1 + \cos(2.094\theta/BW)]$$

$$V_d(\theta) = \sin[1.88\theta/BW]$$

As previously stated, it is necessary to express the error equation (equation 56) as a function of the difference between estimated and actual target location. Such a function allows boresight movement

proportional to this difference. If all angles are considered relative to the boresight axis the radar will estimate a target location, the target may move relative to that location, and the boresight will then move to the estimated position where the whole procedure is repeated over again. In this light, each angular difference between θ and $\hat{\theta}$ is approximately equal to the difference between θ and the boresight (ie $\hat{\theta}$ is approximately equal to zero).

Let $d\theta = \theta - \hat{\theta}$; $\theta = d\theta + \hat{\theta}$; For small $\hat{\theta}$:

$$V_s(\hat{\theta}) = \text{sqrt}(2) \tag{56A}$$

$$V_d(\hat{\theta}) = 1.88\hat{\theta}/BW \text{ which is approximately } 0 \tag{56B}$$

Substitution of equations 56A and 56B into equation 56:

$$e = -A\text{sqrt}(2)\text{sqrt}(E)[\sin(1.88\theta/BW)]$$

$$= -A\text{sqrt}(2E)[\sin(1.88d\theta/BW)] \tag{57}$$

$$d\theta = -BW/1.88\arcsin[e/A\text{sqrt}(2E)] \tag{57A}$$

Equation 57A is the transformation required from an error signal to a step function with amplitude proportional to the required boresight movement for target tracking.

Figure 19 provides a plot of equation 57. Due to the inverse sine function associated with the discriminator curve, all values of error do not map to a unique value of $d\theta$. Limits must be place on $d\theta$ such that within the defined interval, there is approximately a one to one mapping between $d\theta$ and the error signal. For large errors, $d\theta$ can then only move to an established maximum, and no further. This constraint doesn't limit the usefulness of the discriminator curve, because in a realistic

sense the mechanical limitations on how far the boresight can move for large inputs may be the final performance limitation. The limits used for dθ are noted on Figure 19. Equation 57A will be then be used, within the specified limits, to actuate a servo-control system to steer the beam-axis on target.

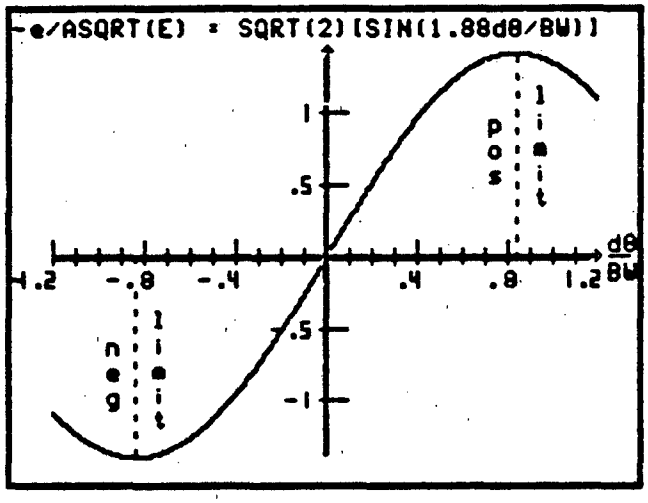

Figure 19. Discriminator Curve

Angle Servo

A tracking radar can be divided into two parts. The first part estimates the position of the target within the resolution cell. The second part centers the resolution cell on the target, typically with the aid of servomechanisms [4:312]. If gear backlash, and other non-linearities, are neglected, the servo system can be modeled as a linear system. Analysis of this linear system can be performed with the aid of Laplace transforms.

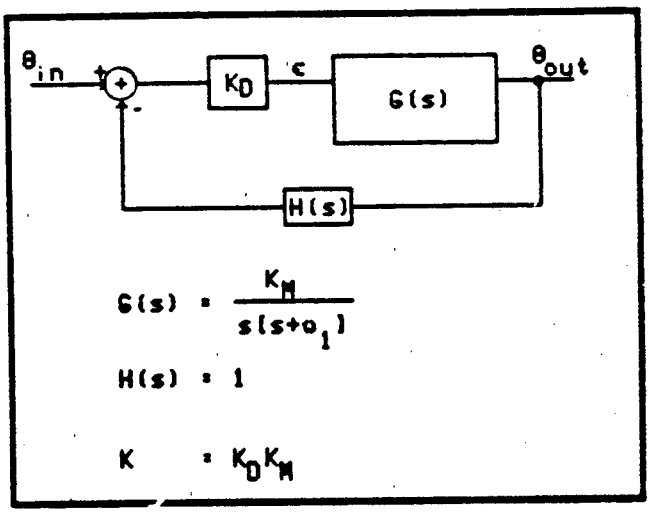

Figure 20. Type I Angle Servo
Source: [4:313]

Figure 20 shows a Type I angle servo and associated parameters. The corner frequency, w_1, is fixed by antenna inertia and motor torque [4:314]. Combining the gains, K_D and K_M of Figure 20, results in an open loop transfer function:

$$G(s) = K/s[s + w_1]$$

Closing the loop, the closed loop transfer function is:

$$\theta_{out}/\theta_{in} = H(s) = G(s)/[1 + G(s)]$$
$$= K/[s(s + w_1) + K]$$
$$= K/[s^2 + sw_1 + K] \qquad (58)$$

The characteristic equation of equation 58 is:

$$s^2 + sw_1 + K = 0 \qquad (58A)$$

The roots of equation 58A determine the time response characteristics of the linear system. Exciting equation 58 by the

Laplace transform of a unit step function (1/s), separating into partial fractions, taking the inverse Laplace transform, and then simplifying the result yields the time domain output response due to a unit step input. For $t \geq 0$, this response can be shown to be:

$$\theta_{out}(t) = 1 - \exp(-\zeta w_n t)[\sin(w_n \text{sqrt}(1 - \zeta^2)t) + \cos(w_n \text{sqrt}(1 - \zeta^2)t) / \text{sqrt}(1 - \zeta^2)] \quad (59)$$

where:

$\zeta = w_1/2\text{sqrt}(K)$ = damping coefficient

$w_n = \text{sqrt}(K)$ = system natural frequency

Since w_1 is fixed by the system elements, desired system performance (ie allowable settling time, steady state error, and percent overshoot in the output response to a unit step function) must be established by the adjustment of the gain parameter, K. Settling time is the time required for the system to damp out all transients. In practice, the settling time is established where the error (ripple about the desired response) is reduced below 5% of the initial error [13:90]. For approximation purposes, the settling time is within 4 time constants of the envelope of the damped sinusoidal oscillation. The ratio of the step response peak value to the steady state settling value is termed percent overshoot. The amount of overshoot allowable depends upon the particular system, and is a function of the damping ratio, but ten percent overshoot is reasonable [13:91]. For the monopulse radar, the constraints for the servo are:

1) The servo response to a pulse must settle to steady state before the arrival of the next pulse. This fixes the settling

time to be less than the PRI, which in turn fixes the rise time of the servo to be less than one-fourth the PRI.

2) To maintain allowable overshoot, the damping ratio must be set to approximately 1/sqrt(2) for optimum second order response to a step input [4:315].

Due to the pulse repetition frequency (PRF) of the radar (100pps), the Type I servo, as described by equation 59, could not settle within the pulse repetition interval (PRI) and a slightly different servo system than the Type I servo must be employed.

Higher gain can be utilized while holding the damping ratio to approximately one-half with the addition of a reciprical corner (phase-lead network) in the system closed loop transfer function of a Type I servo system [14:251]. This phase-lead network eliminates the overshoot problem at high values of gain while reducing the step response rise-time [4:320]. The step response of the improved Type I servo is given in Reference 14 as [14:254]:

$$\theta_{out}(t) = [1 - \exp(-\zeta w_n t)/\mathrm{sqrt}(1 - \zeta^2)]\,\mathrm{sqrt}(1 - w_1/w_n)\,[\cos(w_n\,\mathrm{sqrt}(1 - \zeta^2)t - \phi)] \quad (60)$$

where:

w_1 = antenna corner frequency = 10 rad/sec [4:321] (60A)

w_2 = phase lead network break frequency

w_n = natural frequency = sqrt(K) (60B)

ζ = damping coefficient = $K/(2w_n w_2) + w_1/(2w_n)$ (60C)

ϕ = $\arctan[(1/2\mathrm{sqrt}(1 - \zeta^2))(w_1/w_n + w_n/w_2)]$ (60D)

The rise time is provided as [14:254]:

$$t_r = \frac{(pi) - \arctan[(1/2\,sqrt(1 - \zeta^2))(w_1/w_n + w_n/w_2)] + \emptyset}{w_n\,sqrt(1 - \zeta^2)} \qquad (61)$$

To establish performance at the desired level, the gain (sqrt(K) and the phase-lead break frequency (w_2) were adjusted, by an iterative technique, with constraints of a fixed damping ratio ($\zeta = 0.707$) and a rise time that must be slightly less than one-fourth the PRI ($t_r <$ 2.5msec). For the iteration, the ratio w_n/w_2 was held constant at 1.4 until an acceptable value of rise time was attained, and then w_2 was adjusted until $\zeta = 0.707$. All calculations were performed on a programmable hand-held calculator. The iterative results are:

$$w_n = 900 \text{ rad/sec} \qquad (61A)$$

$$w_2 = 641 \text{ rad/sec} \qquad (61B)$$

From equations 61A and 60B:

$$K = w_n^2 = 810E+3 \qquad (62)$$

Substituting equations 62, 61A, 60A, and 61B into equation 60C:

$$\zeta = 0.707 \qquad (63)$$

Using all of the constants and solving equation 61:

$$t_r = 2.48E-3 \text{ sec} \qquad (64)$$

Solving for the step response (equation 60):

$$\theta_{out}(t) = [1 - \exp(-\zeta w_n t)/sqrt(1 - \zeta^2)]\,sqrt(1 - w_1/w_n)$$
$$[\cos(w_n\,sqrt(1 - \zeta^2)t - \emptyset)]$$
$$= 1 - 1.39\exp(-636t)[\cos(636t - 0.778)]$$

$$\theta_{out}(t) = 1 - 1.39\exp(-636t)[\cos(636t)\cos(.78) - \sin(636t)\sin(.78)]$$

$$= 1 - .99\exp(-636t)[\cos(636t) - .99\sin(636t)]$$

$$= 1 - \exp(-636t)[\cos(636t) - \sin(636t)] \qquad (65)$$

Equation 65 is the output, time domain, servo response characteristic for the monopulse tracker. An amplitude scaled version of equation 65 is used to steer the beam-axis on target for a given error signal input. A plot of equation 65 is provided in Figure 21 for a unit step input.

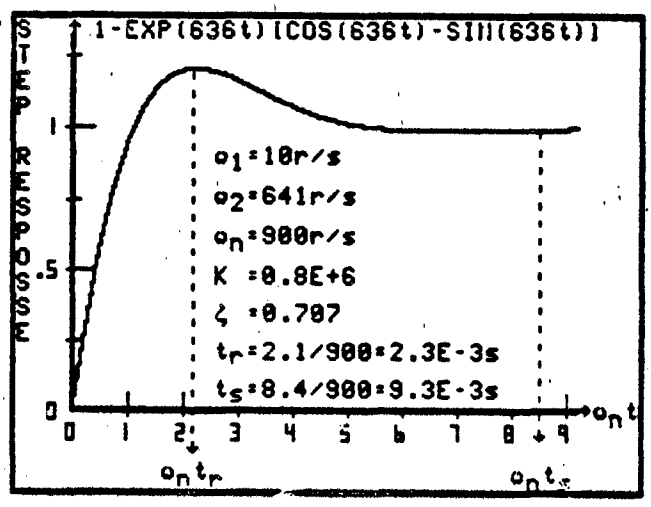

Figure 21. Response of Improved Type I Servo to a Unit Step Function

Moving Target

Several moving target considerations must be accounted for in the radar model. Among the considerations are range and signal-to-noise ratio (SNR) changes because of target movement. Consider linear target translation as shown in Figure 22.

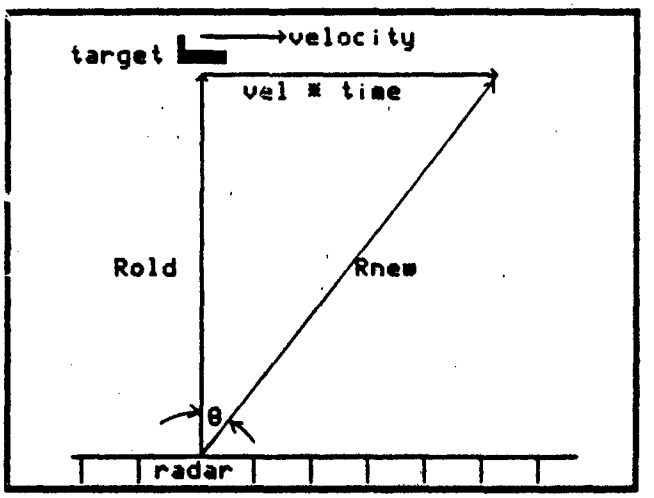

Figure 22. Simple Linear Target Translation

From Figure 22, a simple relationship can be established for the new target range as a function of velocity, time, and previous range.

$$R_{new} = \text{sqrt}[R_{old}^2 + (vt)^2] \tag{66}$$

The range and signal-to-noise ratio are related by the radar range equation [4:29]:

$$R_m^4 = \frac{P_r G_r^2 \sigma_t \lambda^2}{[4(pi)]^3 N_{int} (S/N)_p} \tag{67}$$

where:

R_m = maximum range for a given set of radar parameters

P_r = actual power received by the radar

G_r = antenna gain

λ = wavelength

N_{int} = receiver internal noise power

$(S/N)_p$ = signal-to-noise ration per pulse

σ_t = target cross-section

Solving equation 67 for $(S/N)_p$:

$$(S/N)_p = \frac{P_r G_r^2 \sigma_t \lambda^2}{[4(pi)]^3 N_{int} R_m^4} \qquad (67A)$$

Using equation 67A and after cancellation of like terms, the ratio of new SNR to the previcus SNR is:

$$\frac{SNR(new)}{SNR(old)} = \left[\frac{1/R_m(new)}{1/R_m(old)}\right]^4 = \left[\frac{R_m(old)}{R_m(new)}\right]^4$$

or equivalently:

$$SNR(new) = SNR(old)[R_m(old)/R_m(new)]^4 \qquad (68)$$

where:

$R_m(new) = sqrt[R_{old}^2 + (vt)^2]$

$R_m(old)$ is established by the initial conditions

To establish initial conditions, some constraints must be placed on the simple translation described by Figure 22. Let:

aircraft velocity = 500 knots = 257.4 m/sec

antenna beamwidth = 3 degrees

aircraft travel one-half the beamwidth in 50 pulses

radar PRI = 10E-3 sec

After 50 pulses, the target will travel 50 times the PRI times the velocity, meters. Plugging in the constraints, this translation is 128.7 meters. Figure 23 provides the geometry of the described situation that now allows for initial range computation.

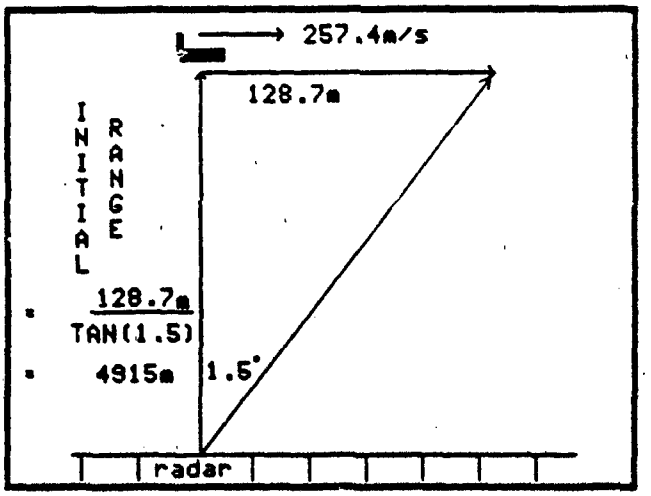

Figure 23. Initial Range Geometry

Solving for the intial target range yields:

$$R_{init} = [128.7 \text{ m}]/\tan(1.5 \text{ degrees}) = 4914.9 \text{ meters}$$

Figure 24 depicts the complete amplitude-comparison monopulse radar development that will be utilized in the radar simulation program that follows.

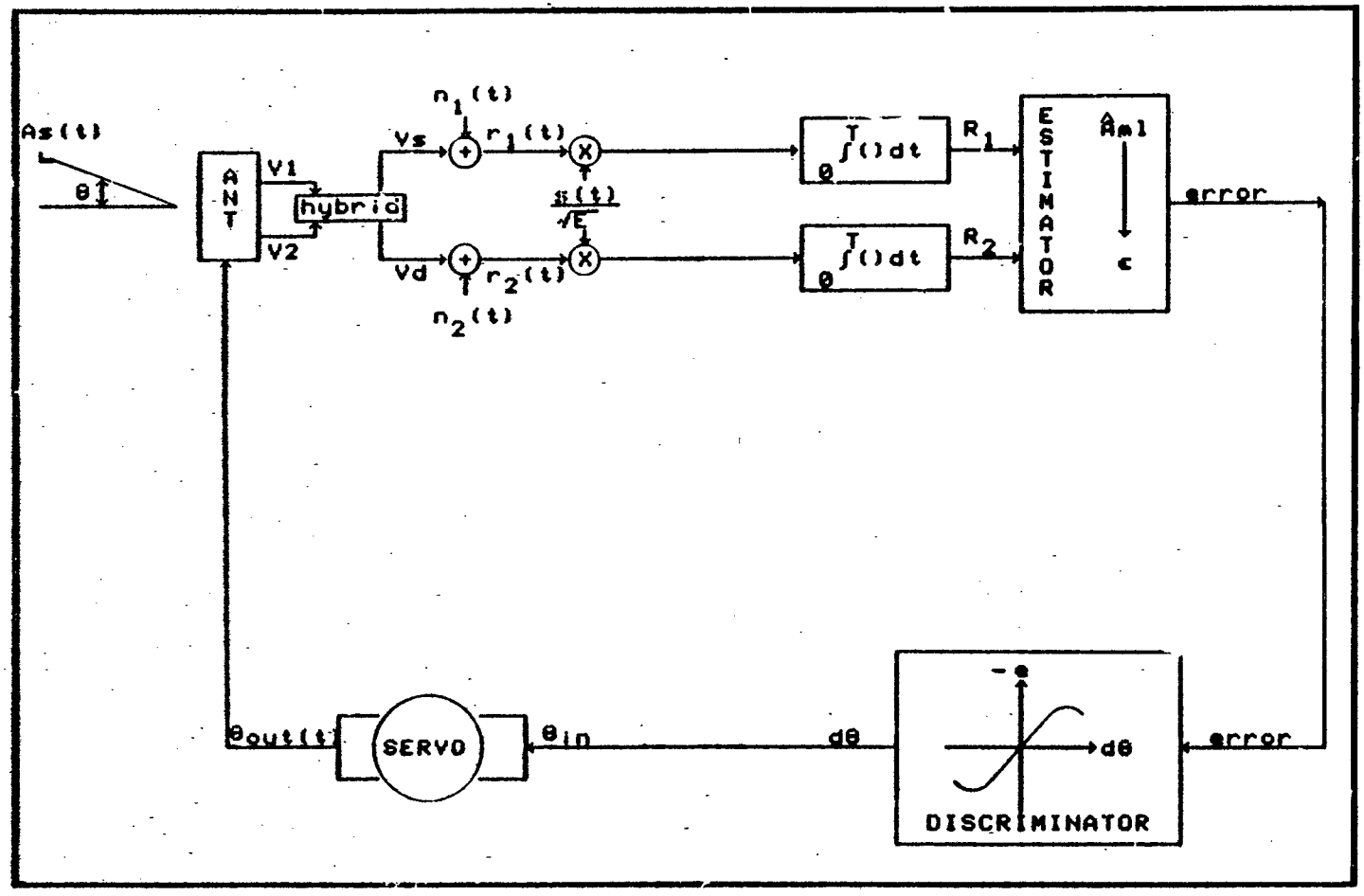

Figure 24. Amplitude-Comparison Monopulse Radar

III. RADAR SIMULATION

Computer Program

To compute the mean-square tracking error and associated performance bounds of the amplitude-comparison monopulse radar, a simulation program was designed for the radar model as developed in the previous chapter. An annotated program listing of the program is provided in Appendix B. The input variables of the simulation allow for choice of a stationary or moving target and desired signal-to-noise ratio (SNR) level. The output parameters are the mean-square tracking error, the Cramer-Rao bound for the conditional variance of the estimated target position, and the Ziv-Zakai lower bound for the estimation error of the target position. Two external International Mathematical and Statistical Libraries (IMSL) routines are called during the program execution. The first routine is a Gaussian random deviate generator, GGNML, that creates random, independent, noise samples. The second routine, MDNOR, computes the area under a Gaussian curve and is utilized to compute the P_e for the Ziv-Zakai bound.

Average tracking error as conducted in the simulation program is developed by observation of the radars ability to maintain the boresight in the vicinity of the target for a finite interval of pulses, and then repeating the experiment many times and averaging the results. The observation interval is established at 50 pulses, and each individual pulse is sampled once. From that one sample, an error signal is generated proportional to the difference between the boresight and

target position. The error signal is then applied to the angle servo to move the boresight in the target direction. Each set of 50 pulses is considered a run. A total of 15 runs are performed and then the error is summed and averaged for each of the 50 pulses over the 15 runs. The simulation program contains four discrete components:

1) An input section
2) A tracking section
3) Computation of performance and bounds
4) An output section

Input

The input is a short section that prompts the user for desired SNR level and whether the target is moving or stationary, establishes the radar and program parameters (pulse width, pulse repetition frequency (PRF), beamwidth, etc.), and positions the target in the beamwidth at the beginning of a run. A few comments about the input section:

1) Two separate 1X50 vectors of zero mean, Gaussian, random variables are established. The two seeds are utilized to insure independent noise samples.
2) Provisions are provided for the testing of 15 signal-to-noise ratio levels in the range from -20 to +35 dB.
3) The only exit for the simulation is the entering of a 0 to the prompt as to whether the target is moving or stationary.
4) Target placement provides for translation to be one-half the beamwidth at the end of the 50th pulse for the initial range, velocity, and PRF provided.

5) Signal amplitude was adjusted while monitoring boresight movement for a stationary target at changing SNR levels. A compromise between the amount of pulses necessary to move to the target and target tracking potential over the established range of SNR levels fixed the amplitude at 5.

A flow chart of the input section is provided in Figure 25.

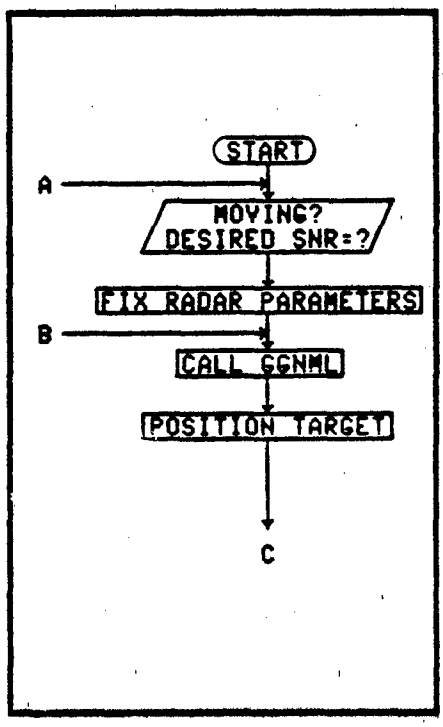

Figure 25: Program Input Section

Tracking

The tracking section moves the boresight to the vicinity of the target for each received pulse. The radar tracks the target for a 50 pulse interval (run). During the run, noise samples are generated from

the stored GGNML vectors, the receiver observations are formulated, an error signal is generated from the observations, and the boresight is moved in a direction to reduce the error. Figure 26 gives a flow chart of the tracking section.

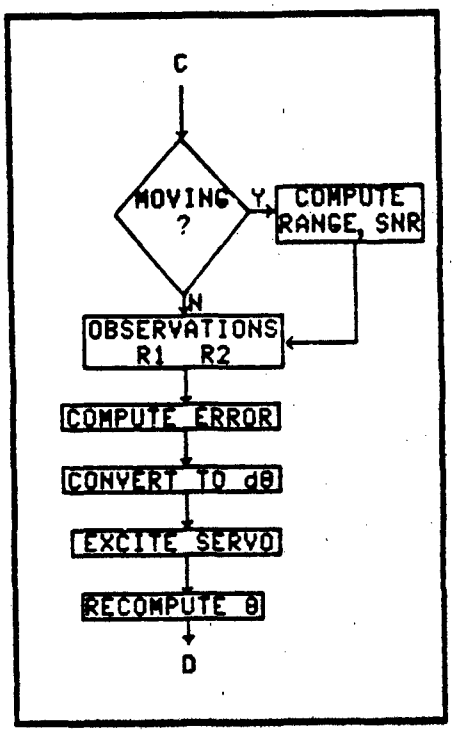

Figure 26. Tracking Section

A few comments about the tracking section:

1) Provisions are provided for the changing range and SNR of moving targets.

2) Large errors are limited to a maximum positive or negative value to model the limitations on how far the boresight can move for large inputs.

3) The time parameter for the servo response is the radar pulse repetition interval (PRI).

Computation of Performance and Bounds

This section computes the squared tracking error, the Cramer-Rao bound, and the Ziv-Zakai bound. The tracking error (angle between boresight and target after observing a pulse) was computed in the last section. Squaring this result (in radians) gives the desired squared error (SE). The Cramer-Rao bound is computed by substitution of the angle θ, after pulse observation, into equation 36. The Ziv-Zakai bound requires the maximum of equation 40C for all target angles from 0 out to θ_p. To find the maximum, the following routine was utilized:

1) Compute the arithmetic mean angle θ_p (θ_p varies from .75 degrees to 1.5 degrees).
2) Segment θ_p into angle intervals with provisions for adjustment of the number of segments (8 segments when θ=0, 16 segments when θ=BW/2) to provide more resolution at large angles (approximately equal step sizes) and then evaluate and store equation 40C for each seqmented interval. The stored values are then sorted, and the maximum is picked off.

Figure 27 provides a flow chart of the described sequence.

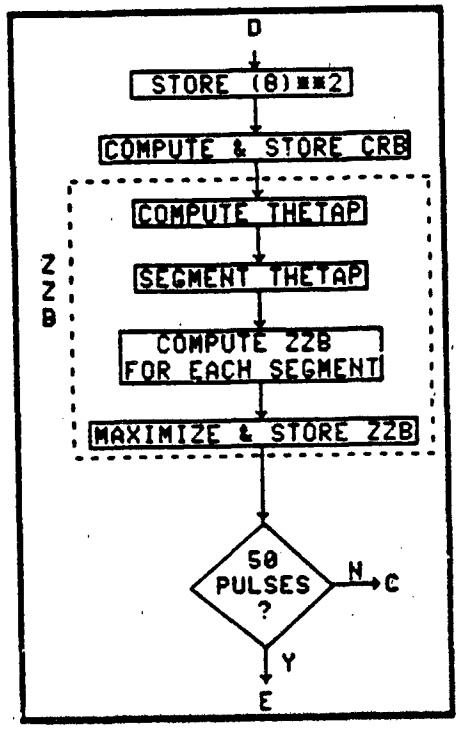

Figure 27. Computation of Performance and Bounds

Output

The output section averages the squared tracking error (SE) and bounds over the 15 runs and prints the three output vectors (the average of the SE will be designated as mean-square error (MSE)). A flow chart of the output section is shown in Figure 28.

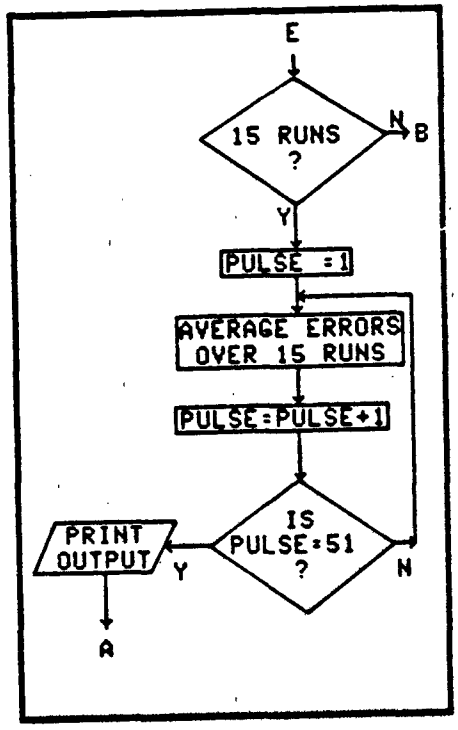

Figure 28. Simulation Output Section

Performance Verification

Stationary Target

The outputs of the simulation program are plotted in the following diagrams. The first plot, Figure 29, shows the general trend of the mean-square tracking error for a stationary target at different SNR levels. As the figure indicates, there is an increasing error for decreasing SNR levels. There is also a general tendency for the mean-square error to be spread over a larger range for the smaller values of SNR. This is a direct indication of the increased difficulty of maintaining the boresight within the vicinity of the target.

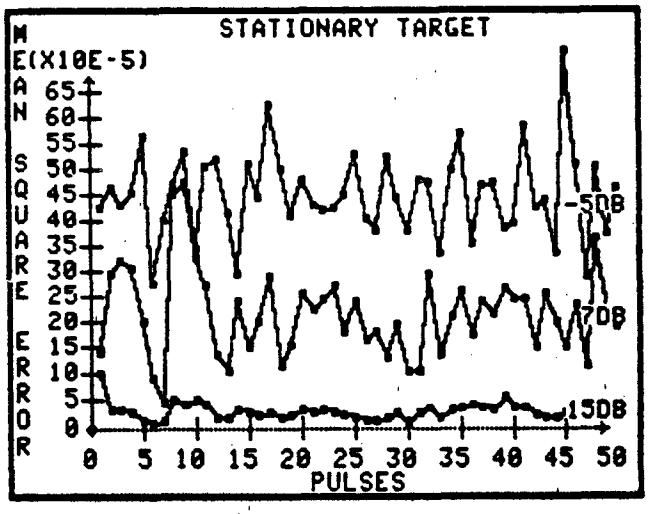

Figure 29. General Trend of the Stationary Target Tracking Error

Figure 30 is a stationary target composite plot for mean-square tracking error (MSE), the Cramer-Rao bound (CRB), and the Ziv-Zakai bound (ZZB) for SNR levels of 15, 7, -5, and -20 dB. As the -5dB plot indicates, the CRB begins to approach the MSE. The -20dB plot clearly shows the CRB exceeding the MSE. Since by definition a lower bound must always be less than or equal to the actual error, the CRB is not useful at low SNR levels. Where the actual bound loses its usefulness depends upon the system parameters, but for the purposes of this simulation, the CRB for a stationary target will only be valid for SNR levels greater than -5dB. In contrast to the CRB, observation of Figure 30 shows the ZZB is tight at the lowest measured SNR level. The -20dB plot shows the ZZB to be approximately constant and no greater than the average MSE excursions.

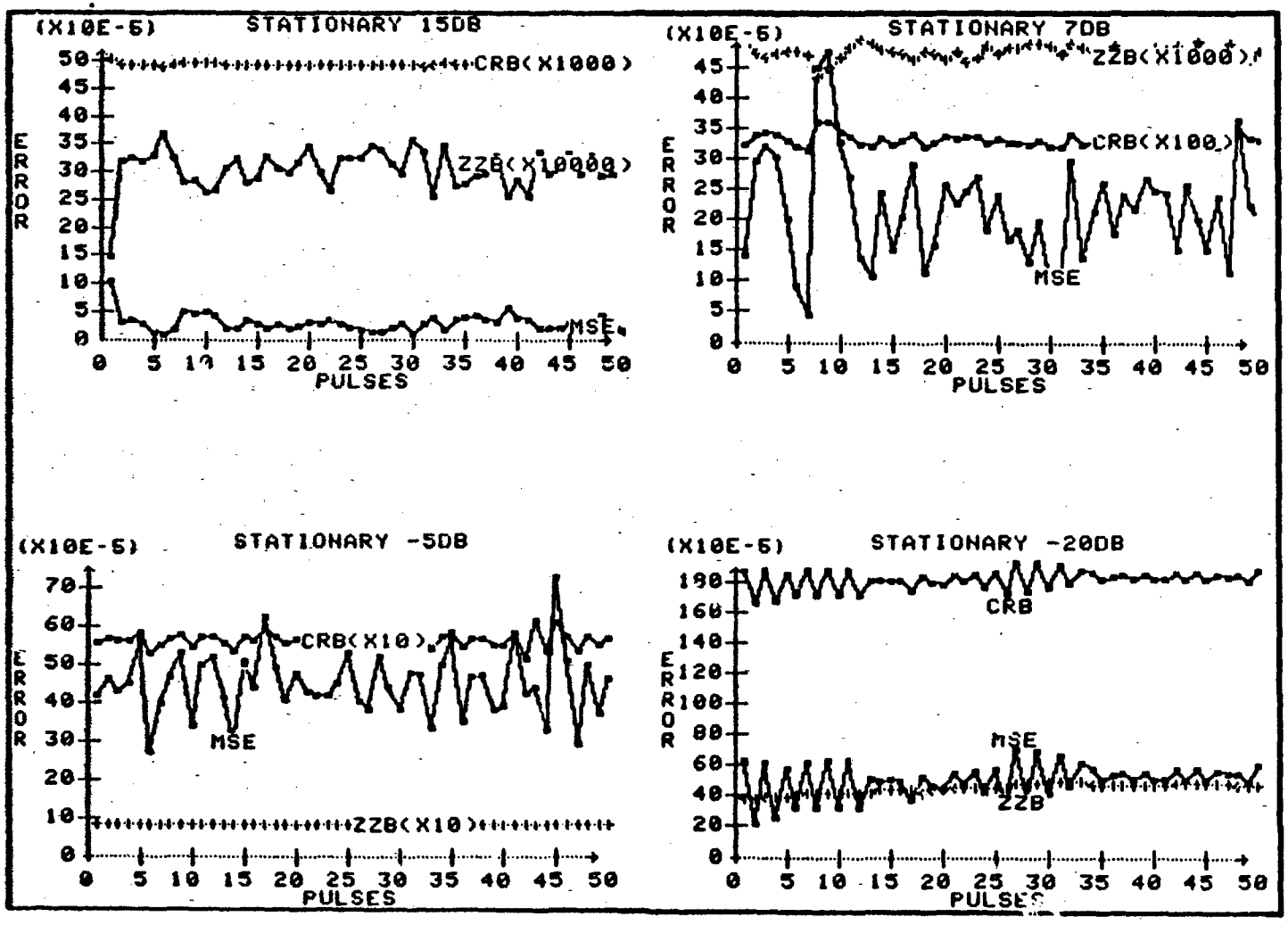

Figure 30. Stationary Target Tracking Error vs Signal-to-Noise Ratio (SNR)

Moving Target

Figure 31 provides the general trend of the mean-square tracking error for a moving target at different SNR levels. As the figure indicates, there is an increasing error for decreasing SNR levels. Further, comparison with the general trend for a stationary target (Figure 29) shows two significant results. The first observation is that for high SNR (15dB) the MSE settles out to the same value as stationary target tracking. The second observation is that for the lower SNR levels, the moving MSE is slightly higher than the corresponding stationary values and the degree of MSE spreading for a moving target is noticeably larger (increased tracking difficulty).

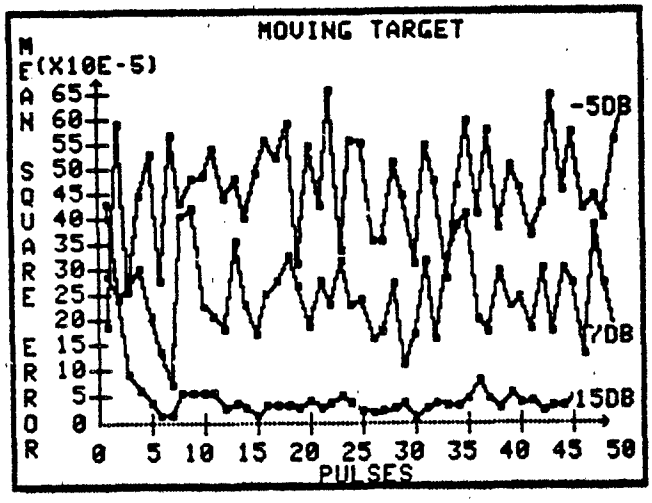

Figure 31. General Trend of the Moving Target Tracking Error

Figure 32 provides the moving target composite plot. As the -20dB plot indicates, the CRB again far exceeds the MSE. In contrast to the CRB, the ZZB is once again tight, approximately constant, and no greater than the average MSE excursions.

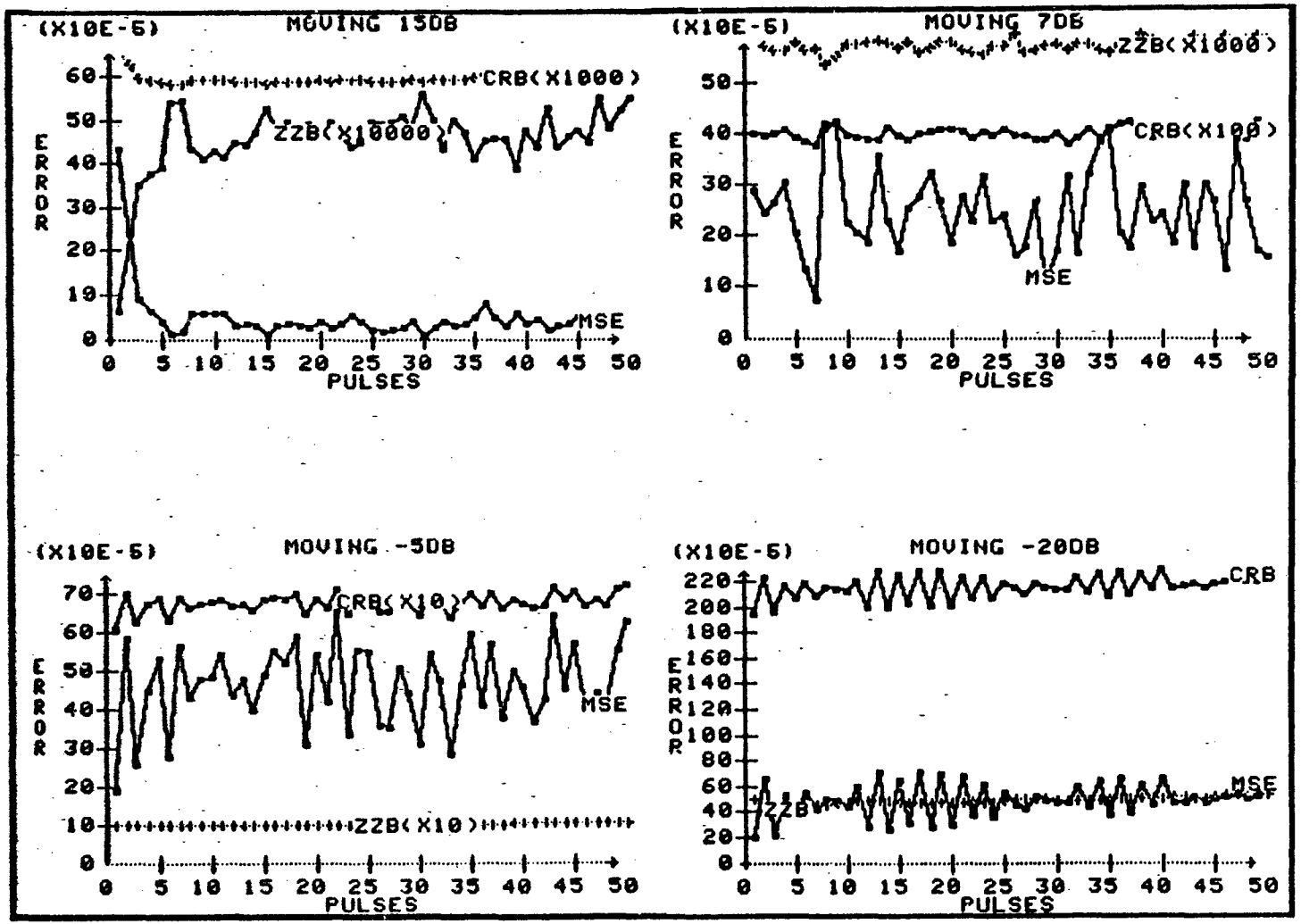

Figure 32. Moving Target Tracking Error vs Signal-to-Noise Ratio (SNR)

IV. TARGET FLUCTUATIONS

The receiver design and analysis up to this point considered only deterministic amplitudes of the received signal. Consequently, the performance estimates, up to this point, are based upon a deterministic amplitude. Of interest in this chapter is what happens to performance estimates, and to target tracking capability, when target fluctuations are present that were not designed for.

To properly account for target cross-sections, the probability density function must be known for the particular type of target that is being tracked. The model of the radar considered thus far has utilized deterministic amplitudes that are proportional to the positive square root of the radar's cross-section. The next extension is to assume that there are many individual point scatterers with each individual scatterer contributing (in an additive fashion) to the cross-section of the target. The probabilistic model of a target in free space with many individual contributers will turn out to appear as one source which has an Exponentially distributed cross-section [15:47]. The corresponding target amplitude will have a Rayleigh distribution. Inherent in this development is the assumption that none of the individual scatterers are much larger than the rest (point targets) and that the cross-section does not change during the sampling interval.

To account for fluctuating cross-sections, the nature of cross-section dependence upon radar viewing angle must be known. The target cross-section is very dependent upon aspect angle, and the aspect angle

changes with time [4:38]. If the cross-section changes significantly over a time of viewing equal to the pulse repetition interval (PRI), the fluctuations are assumed independent from pulse to pulse [15:47]. This paper will consider such fluctuations to be "rapid" fluctuations. If the cross-section changes are significant only over times on the order of a burst of pulses, many times the PRI, the fluctuations are assumed to be independent from scan to scan [15:46]. This paper considers such fluctuations to be "slow" fluctuations. Target fluctuations will be investigated in the following steps:

1) A variable transformation will be performed on an Exponential (cross-section) density to insure that the positive square root (amplitude) will be Rayleigh distributed.
2) The effect of a random amplitude on the receiver estimator will be investigated.
3) The simulation program developed in chapter 3 will be modified to investigate tracking performance with "slow" and "rapid" fluctuations on the signal amplitude.
4) A simple method to reduce the mean-square tracking error for target tracking in the presence of amplitude fluctuations will be proposed and investigated.

Distributions

As previously stated, the probabilistic model of a target in free space with many individual contributors appears to the radar to be one source that has an Exponential distribution on the radar cross-section. Let the cross-section be represented by the random variable X, the

probability density function, pdf, of X is given by [7:45]:

$$f_X(x) = a\exp[-ax] \quad x \geq 0 \tag{69}$$

where:

$a = 1/E(X)$

The units of cross-section are units of power. To convert to a form of positive amplitude, power must be converted to field strength. The appropriate transformation is:

$$Y = +\text{sqrt}(X) \tag{70}$$

The transformation requires the following theorem [7:118]:

> Let X be a continuous random variable and Y = g(X) where g(X) is continuous in X and strictly monotonic. Then
>
> $$f_Y(y) = f_X[g^{-1}(y)]\,|(g^{-1}(y))'| \tag{71}$$
>
> where:
>
> | | denotes absolute value
>
> ( )' denotes the derivitive wrt y

Utilizing equations 69 and 70 in equation 71 and realizing that for X restricted to positive values and Y restricted to the positive square root:

$f_X(x) = a\exp[-ax] \quad x \geq 0$

$g(x) = +\text{sqrt}(X)$

$g^{-1}(y) = y^2$

$|(g^{-1}(y))'| = 2y$

$f_Y(y) = 2ay\exp[-ay^2] \quad y^2 \geq 0$

Let $a = 1/2\sigma^2$

$$f_Y(y) = y/\sigma^2 \exp[-y^2/2\sigma^2] \qquad y \geq 0 \tag{71A}$$

Examination of equation 71A shows that Y is a Rayleigh distributed random variable [11:195]. The general expression for the moments of a Rayleigh distributed random variable are given by [11:148]:

$$E(Y^n) = \text{sqrt}((pi)/2)[1*3*5*.....n\sigma^n]; \quad n \text{ odd} \tag{71B}$$

$$= 2^{.5n}(.5n)!\sigma^n; \quad n \text{ even} \tag{71C}$$

From 71B and 71C, the mean and variance of Y are:

$$E(Y) = \text{sqrt}((pi)/2)\sigma \tag{71D}$$

$$\text{var}(Y) = E(Y^2) - E(Y)^2$$

$$= (2 - (pi)/2)\sigma^2 \tag{71E}$$

The parameter σ of the Rayleigh distributed amplitude will be adjusted in the fade simulation program to force equation 71D to be equal to the deterministic amplitude previously used (A = 5). Note that equation 71E shows that the variance of the random amplitude will change as the parameter σ is changed.

Receiver Structure

Earlier in the estimator development, the necessary condition for the maximum-likelihood estimate was obtained by differentiating the conditional log-likelihood function with respect to the unknown parameter and setting the result equal to zero. This condition was called the log-likelihood equation and was expressed in equation 25A. The inherent assumption was that the conditioning due to the amplitude

and angle of arrival resulted from deterministic, but unknown, parameters. With the amplitude now known to be a Rayleigh random variable, the log-likelihood equation must be expressed in terms of the a priori knowledge of the amplitude distribution. To account for conditioning with a random variable, an average log-likelihood equation can be used to remove the conditioning by averaging over all possible amplitude values [16:310]. Averaging equation 25A:

$$E\{0\} = E\{2r_1V_s\,\mathrm{sqrt}(E)/N_0 - 2A(V_s\,\mathrm{sqrt}(E))^2/N_0 + 2r_2V_d\,\mathrm{sqrt}(E)/N_0 - 2A(V_d\,\mathrm{sqrt}(E))^2/N_0\}$$

$$0 = r_1V_s\,\mathrm{sqrt}(E) - E\{A\}(V_s)^2E + r_2V_d\,\mathrm{sqrt}(E) - E\{A\}(V_d)^2E$$

Solving for $E\{\hat{A}_{ml}\}$:

$$E\{\hat{A}_{ml}\} = \frac{r_1V_s\,\mathrm{sqrt}(E) + r_2V_d\,\mathrm{sqrt}(E)}{(V_s)^2E + (V_d)^2E}$$

$$= \frac{1}{\mathrm{sqrt}(E)}\left[\frac{r_1V_s + r_2V_d}{(V_s)^2 + (V_d)^2}\right] \qquad (72)$$

Comparison of equation 72 and the previously derived estimate for amplitude (equation 26) shows that on the average, the maximum-likelihood estimate for amplitude will be unaffected by its random nature. Recall that the amplitude estimate is later utilized to develop the monopulse error equation, and therefore the receiver structure that develops an error signal of the angle off boresight is, on the average, unaffected by the Rayleigh distributed amplitude. Equation 72 only considers the average value of the amplitude estimate. As shown earlier in equation 71E, the random amplitude will have an associated variance.

This amplitude dispersion will have a direct consequence upon receiver performance for a fixed radar design based upon deterministic amplitudes.

Performance

Rayleigh Amplitude Samples

To investigate the performance aspects of fluctuating targets, it is necessary to modify the simulation program of Appendix B to account for "slow" and "rapid" fluctuations on signal amplitude. The IMSL routine GGWIB provides the necessary tool for developing Rayleigh distributed random samples. GGWIB is a Weibull random deviate generator, and the Rayleigh density is a special case of the more general Weibull density. The general form of the Weibull probability density function, pdf, of the random variable Y is given by [7:229]:

$$f_Y(y) = (A/B)[(y - C)/B]^{(A-1)}\exp[-(y-C)/B]^A; \quad y > C \tag{73}$$

where:

A = shape parameter; A > 0

B = scale parameter; B > 0

C = location parameter

Consider the following parameter values in equation 73:

$$A = 2; \quad B = \sqrt{2}\sigma; \quad C = 0$$

$$f_Y(y) = y/\sigma^2\exp[-y^2/2\sigma^2; \quad y \geq 0 \tag{73A}$$

Comparison of equation 73A with equation 71A shows that with the parameter values as designated, the Weibull density is transformed to

the desired Rayleigh density. This transformation is used in the target fluctuation program to create Rayleigh distributed amplitude samples. An annotated program listing is provided in Appendix C.

Signal-to-Noise Ratio (SNR)

Signal-to-noise ratio as designated up to this point has been equal to the maximum instantaneous signal power to the average noise power. With a fluctuating amplitude, the peak value of signal power is no longer applicable, and an average signal-to-noise ratio must be defined. In general:

$$Egy = \int s^2(t)dt = \int_0^{PW} A^2 dt = A^2 PW$$

An average value of Egy would be:

$$E\{Egy\} = E\{A^2\}PW = 2\sigma^2 PW$$

where:

from 71C, $E\{A^2\} = 2\sigma^2$

Egy represents signal energy

An average signal-to-noise ratio is then:

$$E\{SNR\} = 2E\{Egy\}/N_0 = 4\sigma^2 PW/N_0$$

Taking the ratio of peak to average SNR:

$$SNR_{peak}/SNR_{avg} = Egy/E\{Egy\} = A^2/E\{A^2\}$$

Solving for SNR_{avg}:

$$SNR_{avg} = SNR_{peak} E\{A^2\}/A^2 \qquad (74)$$

In the fluctuating target simulation, the standard deviation of the

Rayleigh distribution was adjusted to provide a mean value of amplitude equal to the nonfluctuating amplitude (A = 5). This was done to provide a performance comparison between fluctuating and nonfluctuating models. Solving equation 71D for the standard deviation required for a mean amplitude equal to 5 yields $\sigma = 3.9894$. From equation 74:

$$\begin{aligned} SNR_{avg} &= SNR_{peak} E\{A^2\}/5^2 \\ &= SNR_{peak} 2\sigma^2/25 \\ &= 1.27 SNR_{peak} \end{aligned} \tag{74A}$$

The result of equation 74A indicates that to maintain the same mean value of amplitude as the nonfluctuating model, 1.27 times as much SNR was required under fading conditions. This is equivalent to adding 1.04 dB more to the signal-to-noise ratio level. Another way of interpreting the result of equation 74A is that for a fixed average SNR, the average and standard deviation of the amplitude would change considerably from the nonfluctuating values of A = 5, and $\sigma = 0$. Because simulation SNR was free, the average SNR could be raised. If there were an associated cost with increasing average SNR, performance may degrade even further than what the results that follow indicate.

Fluctuating Target Simulation

The modifications necessary to include provisions for amplitude fluctuations are simple and straight-forward. The program modification is annoted in Appendix C and includes the following steps:

1) The user is prompted for "slow" or "rapid" fading of the amplitude.

2) The external Weibull deviate generator is called.

3) For "rapid" fading, the pulse amplitude is fluctuated for each of the 50 tracked pulses. For "slow" fading, the amplitude is held constant for each run (50 pulses) but is allowed to fluctuate for successive runs. Each fluctuation is Rayleigh distributed by appropriately transforming the Weibull deviates.

4) The tracking error, and performance bounds, are then computed.

The outputs of the fluctuating target simulation program are plotted for a stationary target in the following diagrams. A stationary target was chosen because as noted in the previous chapter the only difference between stationary and moving targets was a slight increase in the average mean-square error (MSE) value and larger dispersions in the error (increased tracking difficulty). The first plot, Figure 33, shows the general trend of the MSE at high SNR for the nonfluctuating model, the "slowly" fluctuating model, and the "rapidly" fluctuating model. As Figure 33 indicates, even at high SNR the rapidly fluctuating model is very erratic and with a considerably larger tracking error than either the nonfluctuating or "slowly" fluctuating models. This is an indication of the boresight bouncing wildly around the target position rather than settling to a steady-state MSE value. The "slowly" fluctuating model shows a moderate increase in MSE above the nonfluctuating model.

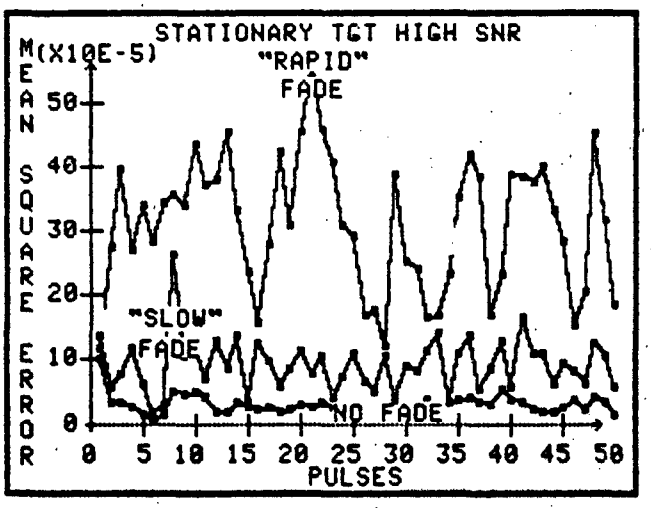

Figure 33. General Trend of Fluctuating Target Tracking Error

Slow Fluctuation

Figure 34 is a stationary target composite plot for the slowly fluctuating target mean-square tracking error (MSE), the Cramer-Rao bound (CRB), and the Ziv-Zakai bound (ZZB) for SNR levels of 15, 7, -7, and -20 dB. Comparison of Figure 34 with the previous nonfluctuating target model shown by Figure 30 demonstrate that:

1) The MSE for "slowly" fading targets shows more degradation at high SNR than for low. This is an indication that when the target tracking environment is poor (low SNR) slow target fades don't contribute significantly to the tracking error.
2) The CRB was largely affected by the presense of target fluctuations.
3) Although the error is greater for all levels of SNR, the radar is still able to track the target.
4) As the -20 dB plot indicates, the ZZB is still tight at the lowest measured SNR level and the CRB has lost its usefulness.

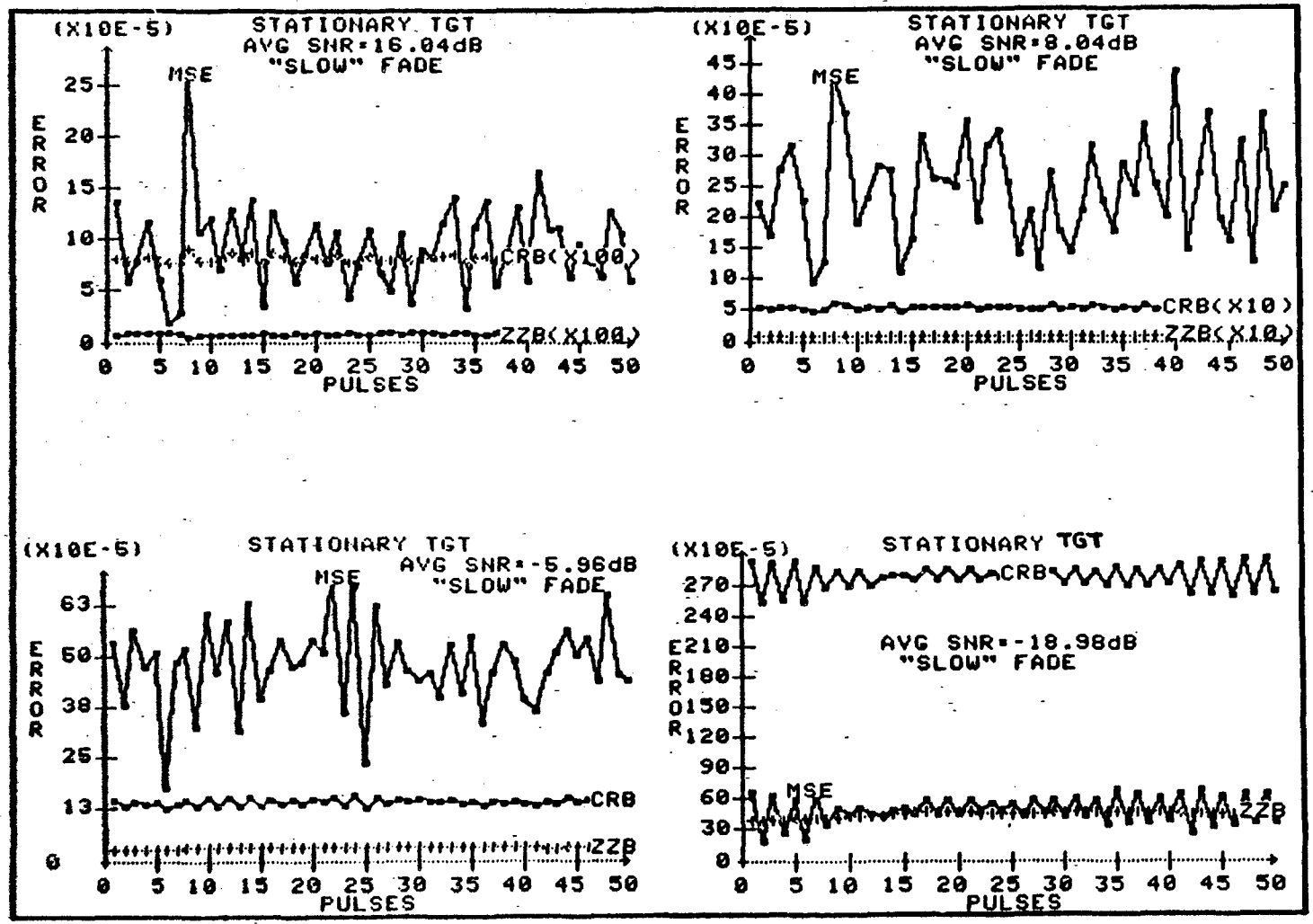

Figure 34. Slowly Fluctuating Target Tracking Error vs SNR Level

Rapid Fluctuation

Figure 35 provides the rapid fluctuation composite plot. Comparison of Figure 35 with the previous nonfluctuating target model shown in Figure 30 demonstrate that:

1) Because of the very large excursions in both MSE and bounds, it is very difficult to track the target in a "rapid" amplitude fluctuation environment.
2) The MSE shows relatively little sensitivity to changing SNR and doesn't really indicate tracking at all.
3) The CRB exceeds the MSE at a higher SNR level.
4) The ZZB shows more variation at low SNR than previously seen, but still proves to be a tight bound at low SNR levels.
5) Characteristic of the "rapid" fluctuating model are deep nulls and peaks from pulse to pulse in the MSE and CRB curves. This indicates that tracking, which should be based upon past history, is independent from pulse to pulse. Clearly this is not a very desireable tracking characteristic.

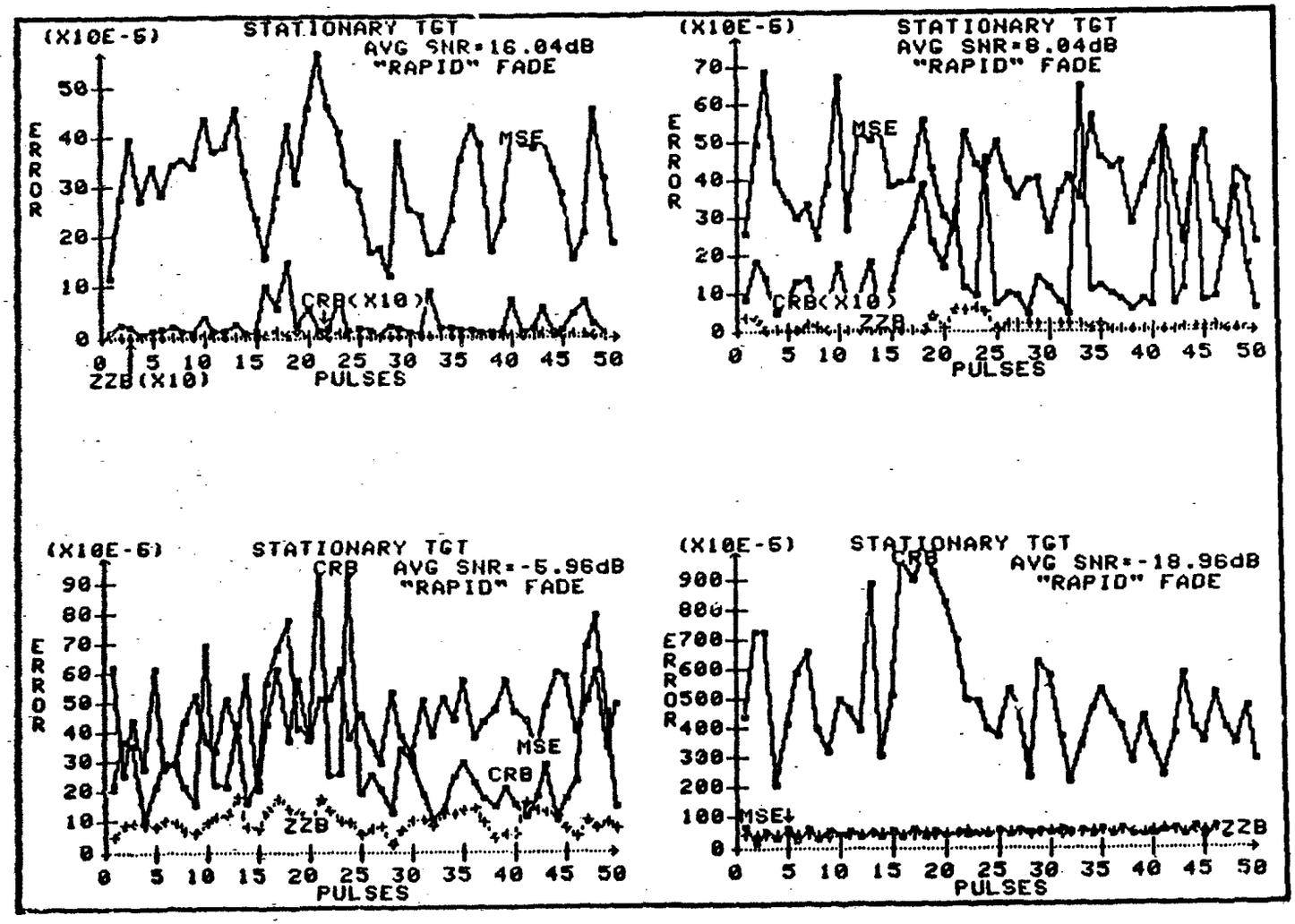

Figure 35. Rapidly Fluctuating Target Tracking Error vs SNR Level

Smoothing Effects

In an attempt to smooth out amplitude fluctuations, the fluctuating target simulation of Appendix C was modified to accumulate the error signal on N pulses before moving the boresight. The annotated smoothing program is provided in Appendix D and includes the following steps:

1) The user is prompted for the number of pulses to be observed before boresight movement (N). N must be a factor of 50 to fit into the previous programming structures.
2) The error for the designated N pulses is then accumulated and averaged. This averaging injects the "smoothing" of amplitude fluctuations over N pulses.
3) The boresight is moved at the end of the Nth pulse based upon the average measured error.
4) Tracking error and bounds are then computed in the same fashion as the previous programs.

Figure 36 provides a plot of the MSE at high SNR for a stationary target with "rapid" amplitude fluctuations and pulse smoothing. Two and five pulses were observed before boresight movement. Also included in Figure 36 is the nonfluctuating MSE. Inspection of the provided plot clearly shows the tendency of smoothing to reduce the tracking error. Further, as the number of pulses increases, the MSE approaches the nonfluctuating model MSE. It is worth noting that this increase in tracking reliability in a fading environment comes at the expense of requiring N pulses to accurately locate target position.

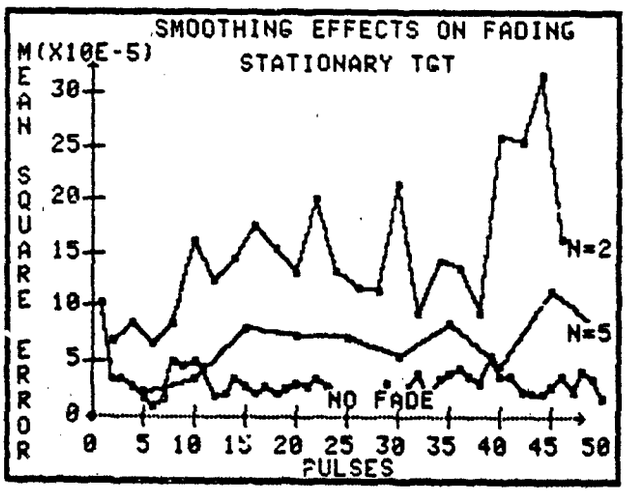

Figure 36. Smoothing Effects

V. MULTIPATH

The last chapter was an extension to the initial model that considered target tracking and performance estimates in the presence of a fluctuating target. This chapter extends the initial model to determine performance estimates and target tracking capability for a more complex channel model, specifically, tracking in the presence of a multipath environment.

As a signal is transmitted from one point to another above an irregular surface with electrical properties differing from that of the propagation medium, reflected signals may be generated. These signals then appear, along with the direct path signal, at the antenna of the radar. The reflected signals are commonly called multipath. The concern of this section is to determine the effects of multipath on azimuth estimation performance.

Rather than a complete spatial diffraction description of specific environmental structures, the attempt will be to develop an adequate model to characterize and simulate the multipath channel. One specific type of multipath will be focused upon, terrain bounce. Figure 37 describes the geometry of the terrain bounce environment. Observe from Figure 37 that the target radiates or reflects signals in all directions and the radar receives a direct signal from the angle θ_t, and a reflected signal from the angle θ_r.

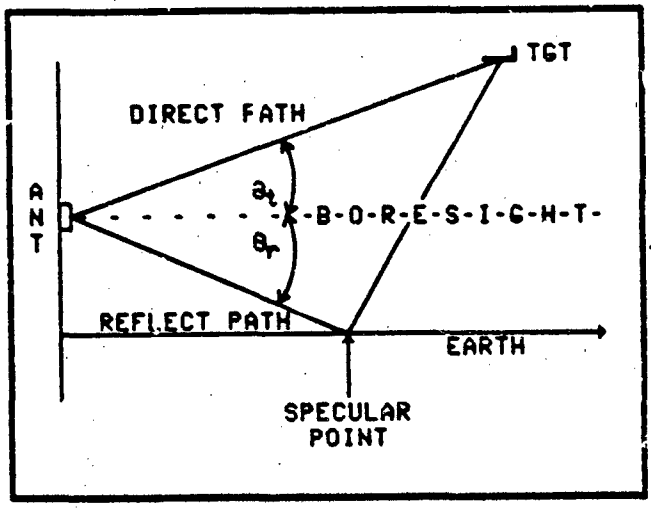

Figure 37. Terrain Bounce Environment
Source [16:688]

The reflected path, or multipath signal return, will depend upon the relection characteristics of the terrain. Strong, well defined returns are called specular multipath and they result from large flat reflecting surfaces. A second kind of multipath, diffuse multipath, occurs due to small surface irregularities and appear to come from all angles instead of a single well defined direction. In this section, both specular and diffuse multipath effects on azimuth estimation performance are of interest. Figure 38 describes the two types of multipath.

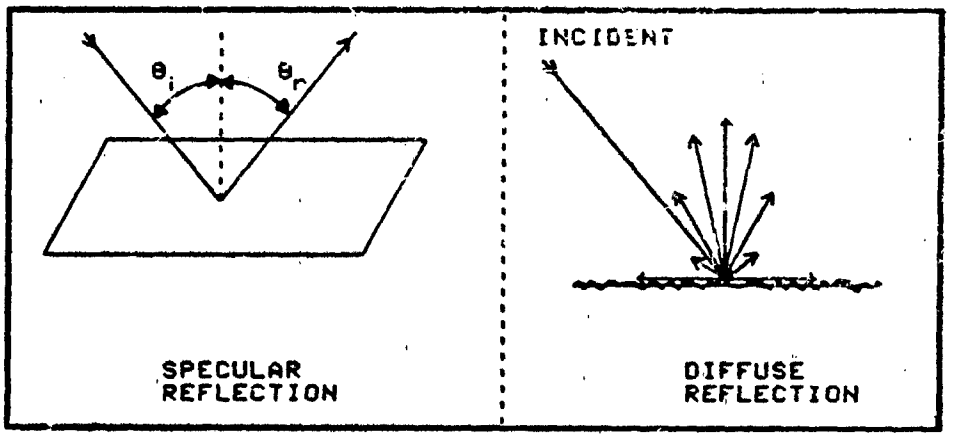

Figure 38. Multipath Types

Tracking in a multipath environment will be investigated in the following steps:

1) A model will be generated for the return signal in the multipath environment.
2) Azimuth estimation performance will be simulated in the presence of specular or diffuse multipath.
3) A method to reduce the tracking error for target tracking in the presence of multipath will be proposed and investigated.

Signal Representation

Characteristic of the multipath medium is the time spread introduced in the signal which is transmitted through the channel [17:455]. A short transmitted pulse over a multipath channel might be received as a train of pulses. Figure 39 shows a possible response to a transmitted pulse through the multipath channel.

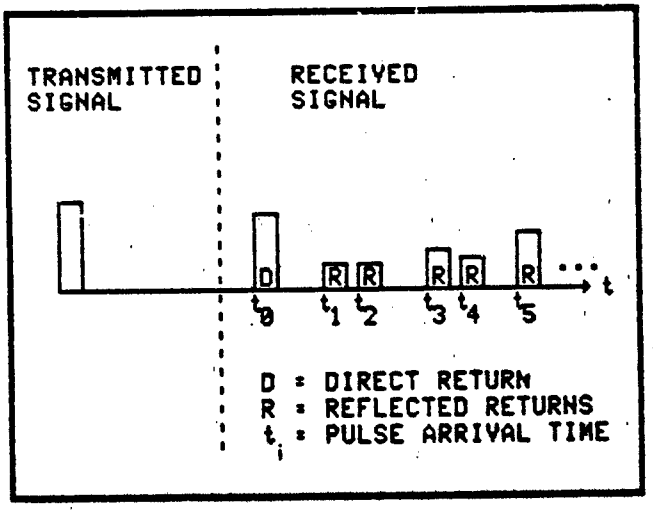

Figure 39. Possible Response to a Transmitted Pulse
Source: [17:455]

Another characteristic inherent to the radar problem is that the received direction of the individual reflected signals will differ from each other and from the direct signal received from the true target location.

The objective of the radar signal model, as developed here, is to obtain a received observation that is a function of both time and azimuthal direction that accounts for the following effects:

1) Direct returns from the target.
2) Reflected returns (either specular or diffuse) from the terrain.
3) The possibility of the superposition of returns (pulse overlap) that may add in a constructive or destructive manner.

General Signal

Most narrowband radar signals can be expressed as [18:298]:

$$s(t) = A(t)\cos[w_0(t) + \phi(t)] \quad (75)$$

where:

w_0 = carrier frequency (rad/sec)

$A(t)$ = instantaneous amplitude

$\phi(t)$ = instantaneous phase

$A(t)$ & $\phi(t)$ vary slowly compared to w_0

To characterize both direct pulses and a random number of reflected pulses with both specular and diffuse multipath environments, equation 75 must be extended to a more general descriptive form of s(t). For any arbitrary pulse, a general representation of the signal received at the radar can be expressed as:

$$s(t) = \mathrm{Re}(u(t)\exp[jw_0(t)]) \quad (75A)$$

where:

$u(t)$ is the complex envelope and can be expressed as

$u(t) = \Gamma A(t)\exp[j\phi(t)]$

Γ = complex reflection coefficient

$A(t)$ = discrete random process that describes the received pulses in both quantity and time of occurance

$\phi(t)$ = arbitrary phase relative to some reference

complex representation is used to allow combining signals

Complex Reflection Coefficient

The reflection coefficient of the terrain surface is generally a complex quantity, $\Gamma = p\exp[j\psi]$. The real part of Γ (p) describes the amplitude changes on reflection, while the imaginary part of Γ (ψ) describes the phase changes on reflection. To assign appropriate magnitude coefficients for the different multipath cases (specular or diffuse), an average value of surface roughness was employed. For average surface roughness, the magnitude (p) of the complex reflection coefficients for both specular and diffuse multipath is approximately 0.4 [16:689]. To assign the appropriate phase after reflection, a phase angle of pi radians applies to a smooth surface with good reflecting properties if the radiation is horizontally (tangential electric fields across a boundary) polarized and the angle of incidence is small (specular) [15:444], and diffuse reflection is assumed uniform in phase from 0 to 2pi. The complex reflection coefficients are:

specular: $\Gamma = 0.4/\underline{pi}$ (76)

diffuse: $\Gamma = 0.4/\underline{\psi}$ (77)

where:

ψ is U(0,2pi)

Envelope A(t)

The changing terrain will cause independent path lengths and a random number of pulse returns, with unique arrival times, at the radar receiving antenna. Defining a random variable A such that A(ζ) equals the total number of returns within the radar pulse repetition interval (PRI), A only takes on discrete integer values 0, 1, 2, ... and the

event (A=k) is the event k returns occur in the time interval [0,PRI). The probability of k returns within the PRI is given by [11:285]:

$$P\{A = k\} = \exp[-\lambda PRI](\lambda PRI)^k/k! \qquad k = 0, 1, \ldots \tag{78}$$

where:

λ = average number of occurances per unit time

Let $a = \lambda PRI$

= rate*time = avg occurances in the time interval [0,PRI)

$$P\{A = k\} = \exp[-a](a)^k/k! \qquad k = 0, 1, \ldots \tag{78A}$$

Equation 78A gives the probabiltiy mass function of A(t), the number of pulse arrivals during the PRI. It describes the Poisson distribution with parameter a [7:171]. At this point, it is necessary to determine the placement of each returned pulse on the time axis. Each time interval between successive pulses will be a random variable and can be termed as the interarrival time for the Poisson distribution. To find the distribution for the interarrival times assume:

1) all events are equiprobable

2) the first event (To) must occur after some time t (ie A(t) = 0 for t < To)

$$P\{A = 0\} = P\{To > t\} = \exp[-a](a)^0/0!$$

$$P\{To > t\} = \exp[-a]$$

$$P\{To < t\} = P\{To \leq t\} = 1 - \exp[-a] \tag{79}$$

The probability distribution of the interarrival times is expressed in equation 79, and its corresponding density function is that of the Exponential distribution [7:209]. The Exponential density function was

previously defined by equation 69. Thus, the envelope of the general representation of the signal received at the radar will take the form of a Poisson distribution with Exponential interarrival times.

Instantaneous Phase

To account for the phase of the complex envelope, all instantaneous phases are assumed as time independent and equiprobable for any phase between 0 and 2pi. This establishes phases as being uniformly distributed from 0 to 2pi and constant. From equation 75A and the preceding development, the received signal at the radar will take on three seperate forms:

1) For the direct pulse; $\psi = 0$, $\phi(t)$ is U(0,2pi), p = 1

$$s(t) = A(t)\cos[w_0(t) + \phi] \quad (80)$$

2) For specular reflections; $\psi = pi$, $\phi(t)$ is U(0,2pi), p = 0.4

$$s(t) = 0.4A(t)\cos[w_0(t) + \phi + pi] \quad (80A)$$

3) For diffuse reflections; ψ=U(0,2pi), $\phi(t)$ is U(0,2pi), p =0.4

$$s(t) = 0.4A(t)\cos[w_0(t) + \phi + \psi] \quad (80B)$$

Observations

The received observations will in general be:

$r(t,\theta) = s(t,\theta)V(\theta) + N(t)$

where:

$s(t,\theta)$ is described by the appropriate form of equation 80

$N(t)$ is additive white Gaussian noise

$V(\theta)$ is an antenna weighting due to the angle of arrival

The model for the antenna weighting, V(θ), will take on one of three seperate forms. The direct pulse will have an angle of arrival as described by Figure 37. The two forms of reflection will have angle of arrivals as described by Figure 38. Therefore:

1) For the direct pulse; θ = target angle of arrival
2) For specular reflections; with respect to the boresight, θ will be uniformly distributed from θ = 0 to θ = BW/2. All reflected pulses will be allowed to form observations.
3) For diffuse reflections; θ will be uniformly distributed from θ = 0 to θ = pi. Only reflected pulses that fall within the interval (0,BW/2) will be allowed to form observations, as all other returns will not get into the received antenna.

Pulse Overlap

With Exponential interarrival times, provisions must be made for the possibility of two pulses arriving into the receiving antenna during the same time interval. Such an occurance would be pulse overlap, and for simplification this investigation considers any ovelap to be complete overlap (ie for any portion of overlap in the pulsewidth, overlap is considered to be of length equal to the pulsewidth) with a resultant received pulse equal to the superposition of the amplitudes and phases of the two affected pulses. This assumption is reasonably valid provided no doppler shifts of either of the two signals occurs (ie the sum of two sinusoids at the same frequency is sinusoidal at the same frequency).

recall $u(t) = A(t)\exp[j\phi(t)]$

in polar form: $u(t) = Um/\underline{\phi m}$ (81)

where:

$Um = |u(t)| = |\Gamma|A(t) = pA(t)$

$\phi m = \underline{/u(t)} = \phi + \psi$

in rectangular form: $u(t) = X + jY$ (81A)

where:

$X = Umcos[\phi m] = Re\{u(t)\}$

$Y = Umsin[\phi m] = Im\{u(t)\}$

Using the rectangular form of u(t) as described by equation 81A, two overlapping pulses can be combined by adding the real and imaginary components of each pulse. For the two overlapping pulses at the same frequency:

$Re\{u_1(t)\} = Um_1cos[\phi m_1]$; $Im\{(u_1(t)\} = Um_1sin[\phi m_1]$

$Re\{u_2(t)\} = Um_2cos[\phi m_2]$; $Im\{(u_2(t)\} = Um_2sin[\phi m_2]$

where:

Um and ϕm are described by equation 81

The resultant magnitude and phase of the sum of the two pulses are:

$$|u(sum)| = sqrt[(Um_1cos(\phi m_1) + Um_2cos(\phi m_2))^2 + (Um_1sin(\phi m_1) + Um_2sin(\phi m_2))^2] \quad (82)$$

$$\underline{/\phi(sum)} = \tan^{-1}\left[\frac{Um_1sin(\phi m_1) + Um_2sin(\phi m_2)}{Um_1cos(\phi m_1) + Um_2cos(\phi m_2)}\right] \quad (83)$$

Only the magnitude (equation 82) is of interest in the combined resultant signal. Equation 82 will be used in the multipath simulation

program to combine two pulses. the resultant signal [$s_r(t)$] is:

$$s_r(t) = \mathrm{Re}(|u(sum)|\underline{/\phi(sum)}\exp[jw_0(t)])$$

$$= |u(sum)|\cos[w_0(t) + \phi(sum)]$$

Performance

To investigate the performance aspects of target tracking in the presence of multipath, extensive modifications are required in the simulation program of Appendix B. Extension to a multipath environment requires a random number of additional pulses that will form observations during each pulse repetition interval (PRI). An annotated listing of the simulation program in the presence of multipath is provided in Appendix E. In the multipath simulation, each received pulse is allowed to form an observation, compute an error estimate, and move the boresight in a direction to reduce the error estimate. To facilitate comparison with previous results the simulation outputs, the tracking error and performance bounds, are again computed after each pulse repetition interval. Both diffuse and specular multipath environments are considered.

Poisson Distributed Returns

As stated in equation 78A, the model used to simulate a random number of arrivals during the PRI is a discrete Poisson distribution on the envelope of the signal, A(t). The IMSL routine GGPOS provides the necessary tool for developing the Poisson deviates. The Poisson parameter, a of equation 78A, is an input parameter, RLAM, of the simulation program of Appendix E, and was adjusted during performance

verification to simulate different multipath environments. Specifically, RLAM values of 0.5, 1.0, and 1.5, respectively, were used to model increasing multipath scenarios. The GGPOS output vector, IR, contains the 50 Poisson deviates describing the number of random reflections for each PRI of the 50 pulse run.

Exponential Interarrival Times

For each of the 50 direct pulses of a run, the Poisson generated number, IR(Pulse) of the simulation, describes the number of reflected pulses. As shown in equation 79, the time of arrival for each of these pulses will be Exponentially distributed. The IMSL routine GGEXP provides the necessary tool for developing pulse interarrival times within the PRI. The number of required Exponential interarrivals, NEX of the simulation, equals the random GGPOS output for a specific PRI. The generated random arrival times, the vector E of the simulation, are sorted to provide sequential arrivals in time. The difference between arrival times, DTIME of the simulation, establishes the response time that the servo-loop will have to each generated observation.

Pulse Overlap

As mentioned earlier, pulses are considered to be totally overlapped if any portion of the pulses overlap (ie if the time difference between pulses is less than twice the pulse width). Both direct and reflected pulses are considered for the possibility of overlap. If overlap occurs, the resultant pulse magnitude is computed by equation 82. Magnitudes are assigned according to the appropriate

reflection coefficient times the pulse amplitude for each considered pulse. All phases are assigned a random, uniform, number between 0 and 2pi. The IMSL routine GGUBS provides the necessary tool for developing uniform deviates from 0 to 1, and an appropriate scaling of the generated vector of uniform deviates yields the desired random phases. Upon computation of the resultant magnitude and phase of two overlapping pulses, a program flag is set which disregards the next incoming pulse and subsequentialy reset in anticipation of another overlap.

Angle of Arrival

The IMSL routine GGUBS also provides provisions for random angle of arrivals (AOA). For diffuse multipath, IR(PULSE) (Poisson number of generated reflections) uniform deviates are generated by an external call to GGUBS and scaled to be uniform from 0 to (pi). Each reflected pulse is then assigned a unique AOA. Likewise, specular multipath uniquely assigns IR(PULSE) AOA's but the deviates are scaled to be uniform from 0 to BW/2. No AOA is assigned to the direct pulse, as it locates the true target location. In the simulation program, no observations are formed for AOA's that are greater than BW/2 (diffuse multipath). Therefore, the difference in the simulation program between specular and diffuse multipath is the AOA's that are allowed to form observations (all for specular; only those between 0 and BW/2 for diffuse).

The outputs of the multipath simulation program are plotted for a stationary target in the following diagrams.

Diffuse Multipath

Figure 40 is a composite plot showing diffuse multipath effects on the mean square tracking error for a stationary target, at high SNR, with varying degrees of multipath (parameter lambda). The upper left corner shows the MSE for the direct pulse with no multipath, and is provided for comparison. Comparison of the plots with parameter lambda unequal to zero with the direct pulse only plot demonstrate that:

1) In all cases, the MSE starts at some large value and settles to a smaller value. This indicates good tracking capability.
2) Although the actual MSE changes from plot to plot (MSE is a random variable), diffuse multipath does not significantly degrade tracking capability. This result verifies a statement made in the literature [19:112] where diffuse multipath is claimed to present no serious limitation in direction finding capabilities.

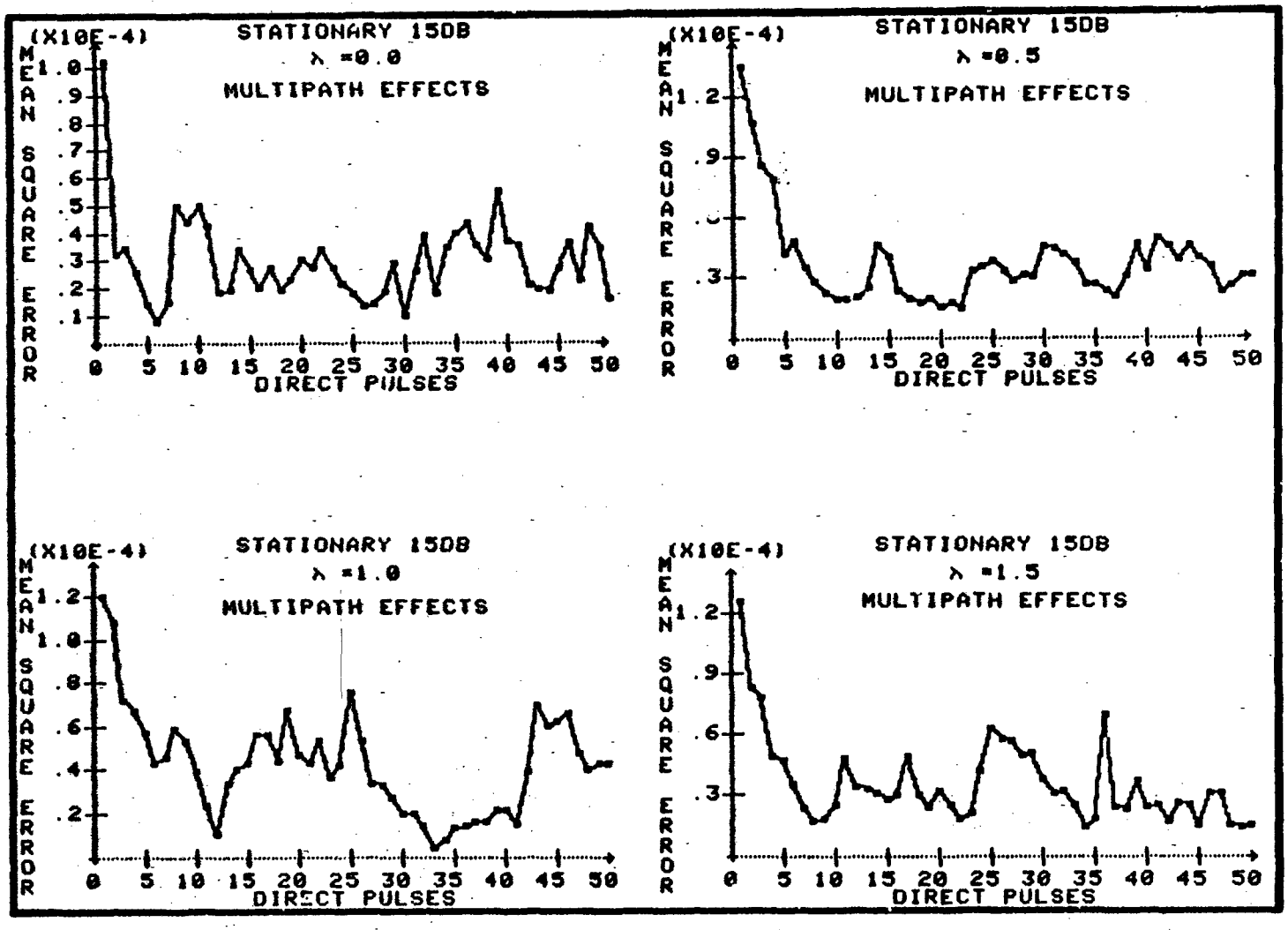

Figure 40. Diffuse Multipath Tracking Error

Specular Multipath

Figure 41 is the corresponding composite plot for specular multipath effects on the MSE for a stationary target, at high SNR, with varying degrees of multipath (parameter lambda). Comparison of the three multipath plots with the direct pulse only plot demonstrate that:

1) Serious monopulse tracking errors occur in the presence of specular multipath.

2) At an average one-half additional pulse per PRI, MSE has increased considerably but still shows tracking capability.

3) At an average of one extra random pulse per PRI, the monopulse radar tracks for approximately 5 pulses and then the tracking error starts to run away. This is an indication of not enough SNR to perform the tracking function (breaklock). Observe that there are a few points where the error dips down, or decreases, in the lambda = 1 plot. The boresight has in these few instances moved to reduce the error and for the purposes of this simulation lambda = 1 is the tracking threshold for the radar.

4) At an average of one and one-half additional pulses per PRI, the tracking error increases monotonically. Clearly, the monopulse tracker has lost its usefulness. The monotonic increase in tracking error may be useful though as a multipath detector [19:61].

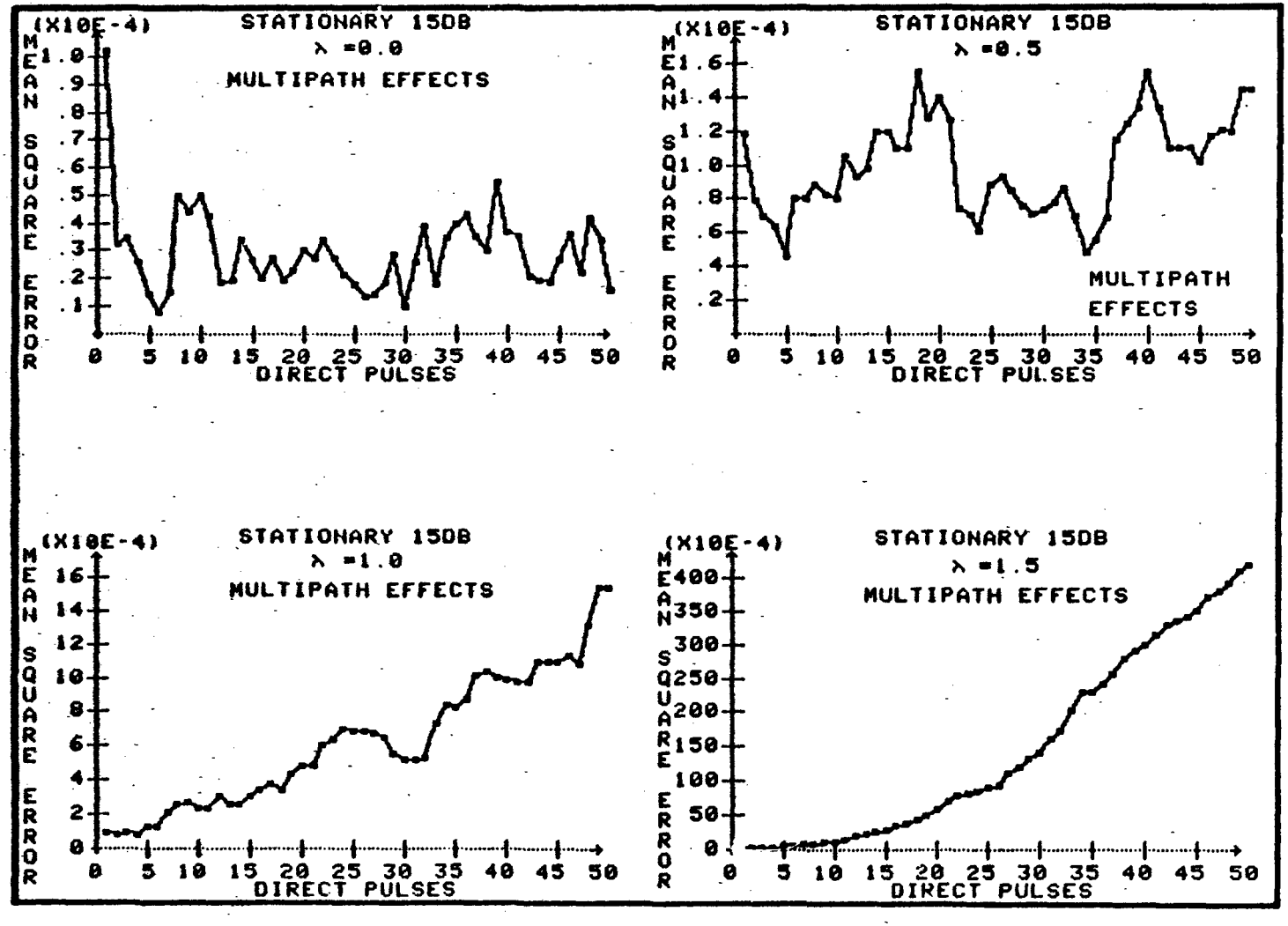

Figure 41. Specular Multipath Tracking Error

Specular Multipath vs SNR

In general, tracking capability was indicated for specular multipath only when the parameter lambda (average number of returns in the PRI) was less than 1.0. Of concern in this section is how tracking and performance bounds are affected by changing SNR levels. Specular multipath was chosen for observation because the previous results indicate that no serious tracking limitations are introduced by diffuse multipath enviornments. Also, since tracking capability was indicated when the parameter lambda was less than 1.0, the Poisson paramenter lambda while monitoring tracking potential will be established as lambda = 0.5. Figure 42 is a stationary target composite plot for MSE, the CRB, and the ZZB for SNR levels of 15, 7, -7, -20 dB in the presence of specular multipath. Comparison of Figure 42 with the previous direct pulse only target model shown by Figure 30 demonstrate that:

1) Even for high SNR levels, the MSE does not settle to a steady state value but is larger at the end of the run (50 pulse) than an average value of MSE. This is indication of a general tendency of the MSE to be increasing in the multipath environment.
2) At -7dB SNR level, although tracking is indicated the monopulse tracker is clearly in a threshold situation where there is not an adequate amount of signal power to determine target direction.
3) Once again the CRB exceeds the MSE at low SNR levels while in contrast the ZZB lower bounds the MSE, even at -20dB.

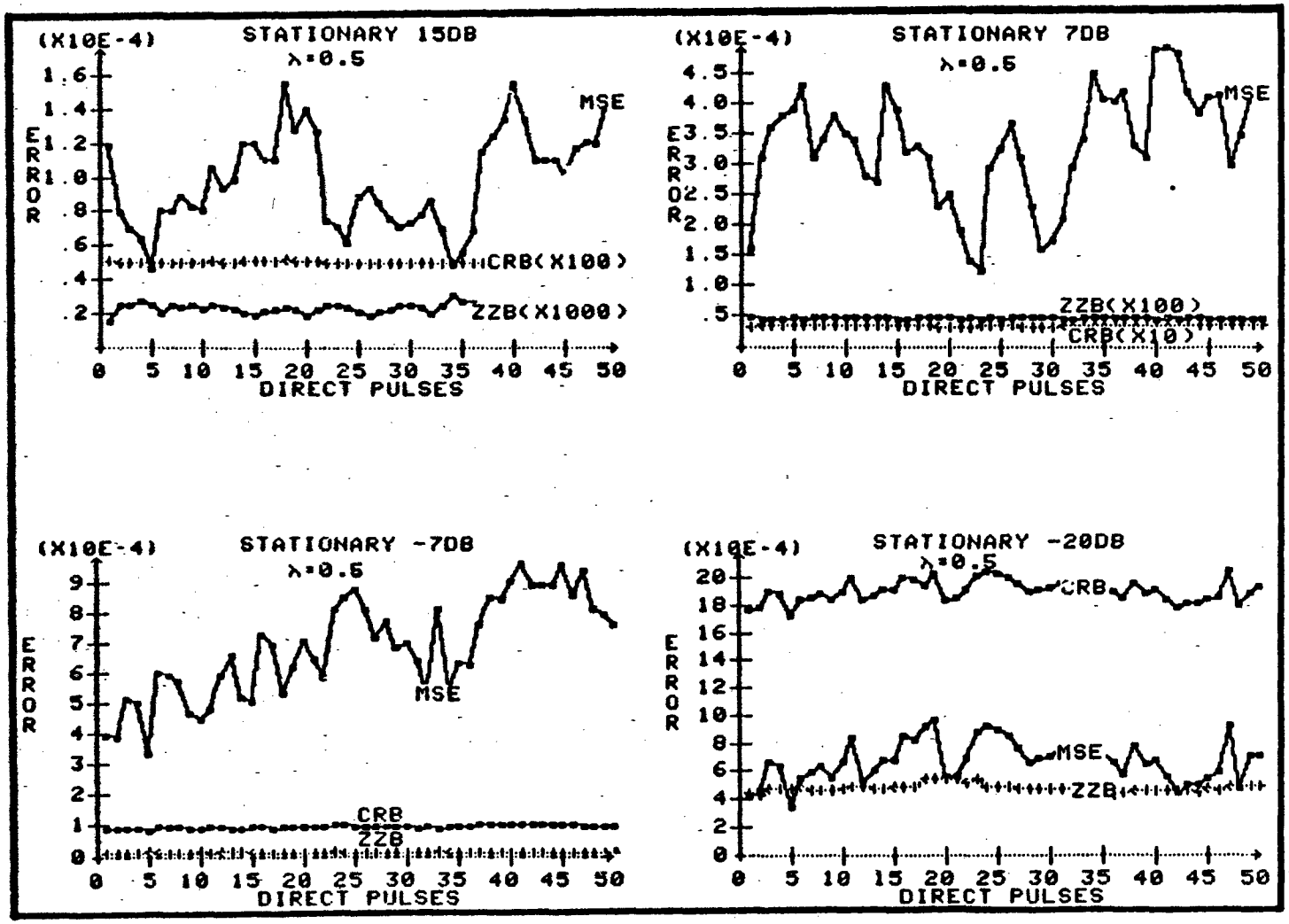

Figure 42. Specular Multipath vs SNR Level

"Lock-out" Technique

In an attempt to overcome the tracking difficulties introduced by a multipath environment, the multipath simulation program of Appendix E was modified to "lock-out" all undesired pulses within the radar's PRI. The "lock-out" technique requires that the first 4 PRI's be tracked in a multipath environment, thus the radar must be able to sacrifice accurate tracking in the first 4 PRI's in favor of increased tracking capability later in the run (the remaining 46 pulses). Additionally, only one target is assumed to be present. Discarding returns (as is done in this technique) with multiple targets present may affect target resolvability.

The objective of the simulation modification is to look at the target returns over the first 4 PRI's and extract some form of periodicity in the target returns. The characteristic focused upon in this investigation was the time of arrival of pulse returns. Recall that the interarrival times are random, Exponentially distributed, over the PRI. The probability that pulse returns would consecutively occur at the same instant in time for the first 4 PRI's and that the pulse would be due to a random terrain reflection is quite small (considered equal to 0 in this simulation). Using this reasoning, the pulse arrival time (arrival time = 0 for the direct pulse, and repetitive) within the first 4 PRI's can be used to "lock-in" the desired target return and "lock-out" all other returns not occuring at the proper time instants.

The annotated "lock-out" program listing is provided in Appendix F and includes the following steps:

1) A (4X25) arrival time storage matrix is initialized to a value of 1.0. The 4 rows represent the first 4 PRI's, and the 25 columns represent the direct pulse and a possible maximum of 24 reflected pulses within each PRI.
2) As the simulation program proceeds, the arrival time for each pulse (direct and reflected) is stored in the proper matrix element position.
3) At the beginning of the 5th PRI, each of the 25 columns are averaged over the 4 PRI's. This column average is then compared to each corresponding column element. If total equality for all elements is obtained with the average and the average is not equal to the initialized value (1.0), or the next PRI, then the equality designates the desired pulse arrival time for "lock-in". All other received pulses will be disregarded in the 5th and subsequent PRI's except the designated "lock" pulse.

Figure 43 provides a plot of the MSE at high SNR for a stationary target in a specular multipath environment and pulse "lock-out" employed. For comparison with the previous performance curves of Figure 41, the same values of lambda (Poisson parameter) were used. Comparison of Figure 43 and Figure 41 show that the "lock-out" technique has removed the tracking degradation introduced by all of the multipath environments. All of the tracking errors settle in to a final value

that is representative of stationary target tracking at high SNR (reference upper left corner plot of Figure 43). It is well worth noting that this increase in tracking reliability in a specular multipath environment is for the particular model of multipath, and form of radar simulation, that has been defined herein and may require extensions to combat an actual multipath situation.

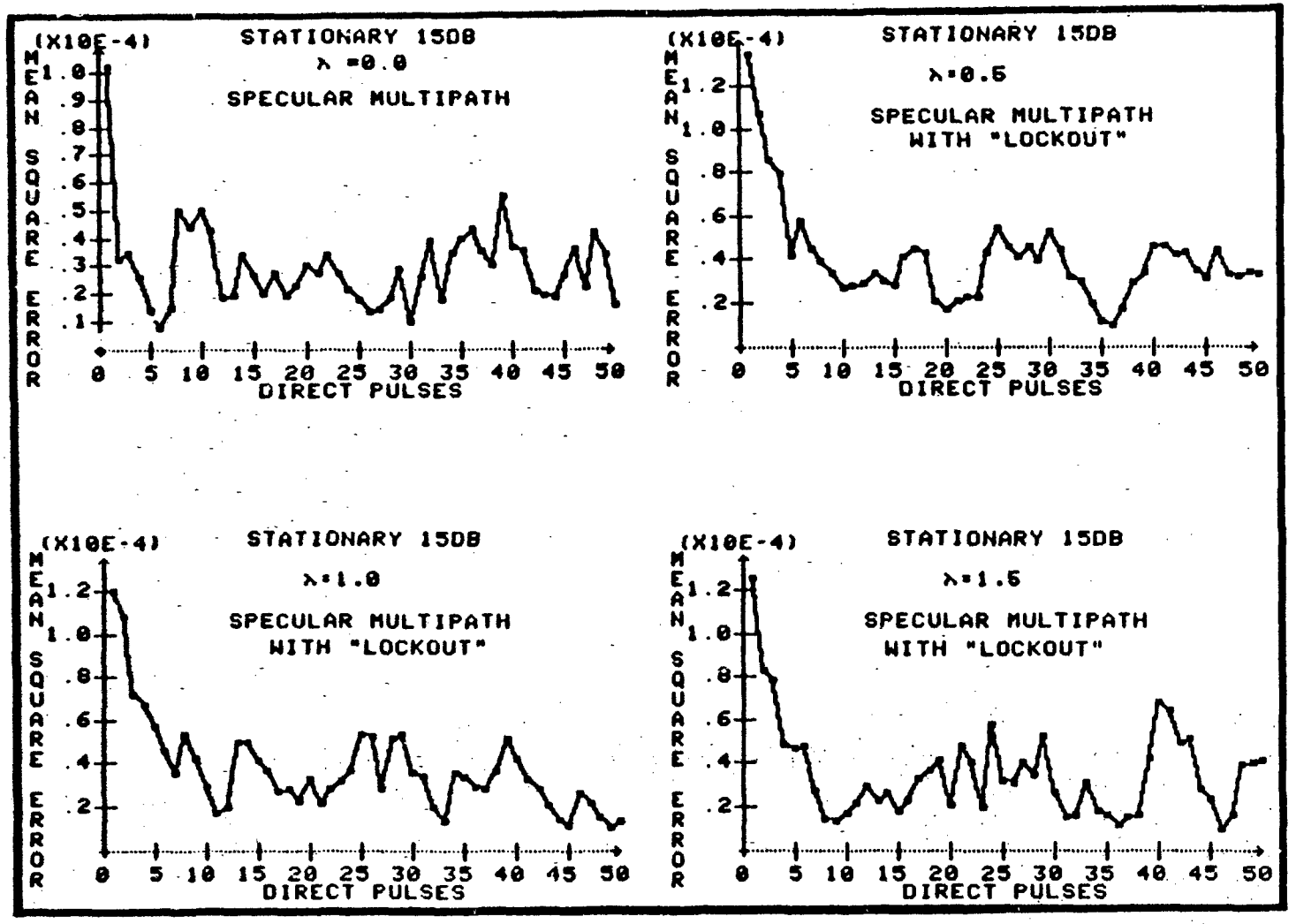

Figure 43. "Lock-out" Effects Upon Multipath Tracking

VI. RESULTS AND RECOMMENDATIONS

This chapter summarizes the previous results for the various tracking capabilities and the applicability of the performance bounds for the amplitude-comparison monopulse radar. In addition, a section is provided for recommendations for continued study along the lines followed by this thesis.

Stationary Target Tracking

As Figure 29 indicates, the general trend of the mean-square tracking error (MSE) for a stationary target is inversely proportional to the available signal-to-noise ratio (SNR) level. There is also a general tendency for the MSE to be spread over a larger range of values for the smaller values of SNR.

Moving Target Tracking

Figure 31 indicated two significant results for the MSE for a moving target relative to the previous stationary target tracking trends:

1) For high SNR, the moving target MSE settles out to the same value as for stationary target tracking.
2) For low SNR levels, the moving target MSE is slightly higher than the corresponding stationary values, and the MSE spreading is noticeably larger (increased tracking difficulty).

Cramer-Rao Bound

Present in all of the obtained results is the inability of the Cramer-Rao bound (CRB) to predict tracking capability in low SNR environments. Performance results indicate that the applicability of the CRB as a performance indicator is limited to prior knowledge that the MSE exceeds the bound (ie the SNR is above a threshold situation). Further, as Figure 35 indicates, the CRB is extremely susceptable to target amplitude fluctuations (producing a much higher error with correspondingly larger valleys and peaks in a fluctuating target environment).

Ziv-Zakai Bound

The Ziv-Zakai bound (ZZB) in all observed tracking environments was approximately constant and no greater than the average MSE excursions. For high SNR levels, a large magnitude difference exists between the ZZB and MSE (See Figure 30). Thus, the ZZB is more applicable at lower SNR levels. For low SNR levels, the ZZB proved to be very tight to actual performance values in that the bound always existed at, or slightly below, the arithmetic mean of the MSE excursions (see the -20dB SNR level plots of Figures 30, 32, 34, 35, and 42). Thus, the ZZB was found to be a very good performance indicator for low SNR levels.

Target Fluctuations

Significant results of tracking in a fluctuating target amplitude environment are:

1) The maximum-likelihood receiver structure was unaffected by the nature of the fluctuating target amplitude.
2) "Slow" amplitude dispersions only moderately degraded tracking capability (see Figure 34).
3) "Rapid" amplitude dispersions imposed serious tracking limitations (tracking degradation) upon a fixed radar design based upon deterministic amplitudes (see Figure 35).

Multipath

Significant results of tracking in a multipath environment are:

1) Diffuse multipath presents no serious limitations in target tracking capabilities (see Figure 40).
2) Serious tracking errors occur in the presence of specular multipath (see Figures 41 and 42).

Recommendations

In the way of recommendations for continued study along the lines established by this thesis, the following studies are proposed:

Simultaneous Parameter Estimation

One of the possible extensions of the work presented here would be to include another radar parameter in the estimation process and investigate the multiple parameter ambiguity fuction. Possible candidates for the study might be the signal delay or doppler.

Two-Dimensional Radar

Appendix A provides the work done in this thesis on a two-dimensional target tracking radar. The mapping from probability space to a moment space (in the estimation error) was accomplished via a two-dimensional generalization of the Tchebycheff inequality. The method utilized to come up with a two-dimensional mean-square estimation error requires close scrutiny as to its correctness and applicability. Further, the error probabilities for both estimation and detection as calculated in this development require exhaustive integrations of standard normal density functions at low SNR levels. The efficiency of the computational algorithm could be considerably improved with a Q function "look-up" table rather than using computer integrations. Such an improvement would allow runs at lower SNR levels than were possible in this investigation and would directly provide indication about the applicability of this form of the ZZB to the two-dimensional tracker.

Simultaneous Multipath Components

This thesis focused in on either specular or diffuse multipath environments when in a practical tracking environment both components would be present simultaneously. It is well worth recommending an investigation that considers both multipath components present and their resulting consequences upon receiver performance. The idea would be to establish a probabilistic description of the received envelope by allowing a constant specular component added to a Gaussian distributed diffuse component (Central Limit Theorem applied to many diffuse scatterers). The constant plus a Gaussian should result in a Rician

envelope distribution. It would be interesting to monitor receiver performance while varying the specular component of multipath [ie varying the specular component from zero (diffuse component only), to an intermediate value (combined specular and diffuse components), to a large value (specular component only)].

Appendix A

TWO-DIMENSIONAL RADAR MODEL

Introduction

In its simplist form, amplitude-comparison monopulse radar determines target direction by comparing a single pair of signals received on two antenna patterns simultaneously. This is sufficient to determine the target angle of arrival in a single plane (azimuth or elevation). This simple form of monopulse is the radar model thus far developed. Three dimensional tracking, however, requires comparison of two pairs of signals in orthogonal planes (usually one in azimuth and the other in elevation).

The two-dimensional development that follows provides only significant features and results. Extensive derivations are omitted as much of the radar development parallels previous work, and a rigorous development is lengthy. The scope of this appendix is to cover the two-dimensional radar model with sufficient detail to demonstrate the work done in this thesis, and to provide an avenue for subsequent studies.

Antenna Functions

Figure 44 shows the monopulse antenna signal processing circuitry for a four horn monopulse radar. Amplitude-comparison monopulse uses the amplitude difference between two adjacent horn voltages to generate error signals [20:46].

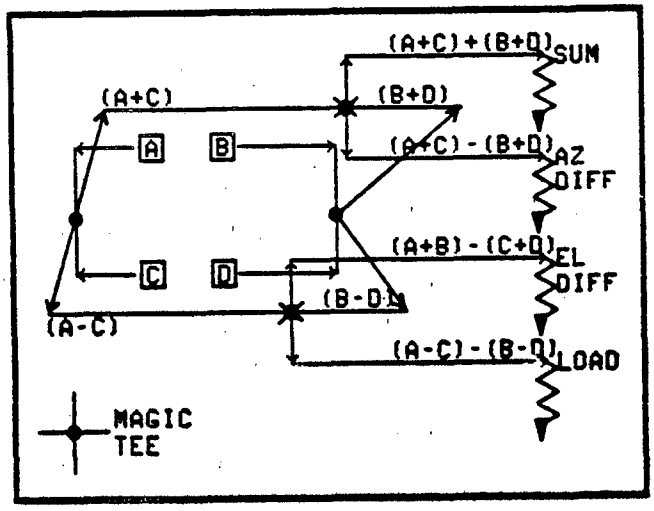

Figure 44. Monopulse Antenna Signal Processing Circuitry
Source: [20:46]

Inspection of Figure 44 indicates that only three combinations use adjacent horns (A + D or C + B are diagonal combinations). The three adjacent horn combinations produce a sum channel, an azimuthal difference channel, and an elevation difference channel. The diagonal combination is not used, and is simply routed to a load.

Reference 4 derived the three necessary antenna patterns required for further processing of the two-dimensional monopulse receive signals. Gaussian beams squinted off the tracking axis in the two-orthogonal coordinate directions, azimuth and elevation, were used. The Gaussian pattern is almost identical (out to one-half the beamwidth) to the Sa() pattern resulting from the uniformly illuminated rectangular aperature that was developed in the second chapter [4:268]. Since all angles of interest are within one-half the beamwidth, the squinted Gaussian beams can be successfuly applied to this two-dimensional development. The

derived sum and difference patterns of squinted beams (for small angular displacements) are [4:306]:

$$V_s = 2g_0$$

$$V_e = 2g_0 k_m \theta/BW$$

$$V_a = 2g_0 k_m \phi/BW$$

where:

V_s = sum voltage pattern

V_e = elevation difference voltage pattern

V_a = azimuthal difference voltage pattern

k_m = $2\sqrt{2}\ln(2)\theta_s/BW$; $2\sqrt{2}\ln(2)\phi_s/BW$

= 0.98 if $\theta_s = BW/2$; $\phi_s = BW/2$

g_0 = pattern gain constant

Let: $g_0 = 0.5$

$$V_s = 1 \tag{84}$$

$$V_e = \theta/BW \tag{84A}$$

$$V_a = \phi/BW \tag{84B}$$

The three forms of equation 84 are the desired antenna function approximations for the two-dimensional monopulse radar model.

Estimation Model

Observation Formulation

For the specific case of additive white noise, the received waveforms for the sum and difference channels are:

$$r_1(t) = As(t)V_s + n_1(t) \tag{85}$$

$$r_2(t) = As(t)V_e + n_2(t) \tag{85A}$$

$$r_3(t) = As(t)V_a + n_3(t) \tag{85B}$$

Applying the Gram-Schmidt orthogonalization procedure as in the Chapter 2 development, the waveforms can be replaced by finite dimensional vectors. Characterization of the received observation by way of a joint density function can then be made. The vector observations are:

$$R_1 = AV_s\,\mathrm{sqrt}(E) + N_1 \tag{86}$$

$$R_2 = AV_e\,\mathrm{sqrt}(E) + N_2 \tag{86A}$$

$$R_3 = AV_a\,\mathrm{sqrt}(E) + N_3 \tag{86B}$$

The means and variances of the three observation vectors are:

$$E\{R_1/A,\theta,\phi\} = A\,\mathrm{sqrt}(E)V_s;\quad \mathrm{var}\{R_1/A,\theta,\phi\} = N_0/2 \tag{87}$$

$$E\{R_2/A,\theta,\phi\} = A\,\mathrm{sqrt}(E)V_e;\quad \mathrm{var}\{R_2/A,\theta,\phi\} = N_0/2 \tag{87A}$$

$$E\{R_3/A,\theta,\phi\} = A\,\mathrm{sqrt}(E)V_a;\quad \mathrm{var}\{R_3/A,\theta,\phi\} = N_0/2 \tag{87B}$$

The conditional covariances are:

$$\mathrm{cov}\{R_1,R_2/A,\theta,\phi\} = \mathrm{cov}\{R_1,R_3/A,\theta,\phi\} = \mathrm{cov}\{R_2,R_3/A,\theta,\phi\} = 0 \tag{88}$$

Equation 88 shows the vector observations to be uncorrelated. Further, with Gaussian noise assumed and linear operations to develop the vector observations, R_1 R_2 and R_3 will be Gaussian and uncorrelated, therefore independent. The receiver observation is then the joint density of three independent Gaussian random variables. Independence implies that the joint density of R_1, R_2, and R_3 is the product of the

marginal distributions. With Gaussian distributions and statistics given by equations 87, 87A, 87B, and 88 the joint conditional density of the receiver observation is:

$$p_R(r/A,\theta,\phi) = p(r_1/A,\theta,\phi)p(r_2/A,\theta,\phi)p(r_3/A,\theta,\phi)$$
$$= [(pi)N_0]^{1.5}\exp(-1/N_0[(r_1 - AV_s\,sqrt(E))^2 + (r_2 - AV_e\,sqrt(E))^2 + (r_3 - AV_a\,sqrt(E))^2]) \quad (89)$$

Maximum-Likelihood (ml) estimation

The log-likelihood equation for a parameter A was given in equation 25 of Chapter 2. The extension to a three dimensional conditional density described by the vectors R_1, R_2, and R_3 is straightforward. Applying the procedure described by equation 25 to the joint conditional density of equation 89 and solving for the unknown parameter A yields:

$$\hat{A}_{ml} = \frac{1}{sqrt(E)} \left[\frac{r_1V_s + r_2V_e + r_3V_a}{(V_s)^2 + (V_e)^2 + (V_a)^2}\right] \quad (90)$$

Taking the expected value of equation 90 produces the mean value of the amplitude estimate. Performing the expectation, the amplitude estimate, as in the one-dimensional case, proves to be unbiased.

Applying the maximum-likelihood procedure of equation 25 to an estimate of the elevation angle off boresight ($\hat{\theta}$) and substitution of the amplitude estimate of equation 90 into the result yields the elevation error equation. The elevation error (e_e) is:

$$e_e = r_1\hat{\theta}_{ml}BW + r_3\hat{\theta}_{ml}\phi - r_2[BW^2 + \phi^2] \quad (91)$$

where:

The antenna functions of equations 84, 84A, and 84B are used in the development for elevation error

Taking the expected value of equation 91 produces the average elevation error. Performing the expectation operation, the average error value is found to be equal to zero when $\hat{\theta}_{ml}$ equals the parameter θ. Assuming zero error, equation 91 can be solved uniquely for the elevation maximum-likelihood estimate. The result is:

$$\hat{\theta}_{ml} = \frac{r_2[BW^2 + \phi^2]}{r_1BW + r_3\phi} \quad (92)$$

Manipulation of equation 92 and taking the expectation of both sides yields the average estimate value of the elevation angle estimate. Performing the expectation operation, the maximum-likelihood estimate for the elevation angle off boresight is found to be unbiased (ie the average value of the estimate equals θ). An operation similar to that for the equation 91 development performed for the azimuthal angle off boresight produces the azimuth error equation (e_a), and an unbiased maximum-likelihood estimate for the azimuthal angle off boresight.

$$e_a = r_1\hat{\phi}_{ml}BW + r_2\hat{\phi}_{ml}\theta - r_3[BW^2 + \theta^2] \quad (93)$$

$$\hat{\phi}_{ml} = \frac{r_3[BW^2 + \theta^2]}{r_1BW + r_2\theta} \quad (94)$$

When joint azimuth and elevation angle estimates are made, the

estimates are, in general, coupled. Inspection of equations 92 and 94 show that both estimates have a scaled Gaussian numerator and a Gaussian denominator (sum of scaled Gaussian random variables). The ratio of jointly independent Gaussian random variables has a Cauchy density centered at zero [11:198]. The Cauchy density is like the Gaussian density, but the tails are off-axis. Moments do not exist for the Cauchy density, thus making it difficult to determine the coupling coefficient for two Cauchy random variables (azimuth and elevation estimates). This investigation will make use of reference 21 which generated combined azimuth and elevation maximum-likelihood estimates in a white noise environment. Reference 21 shows that for a narrow-band signal the maximum-likelihood elevation and azimuth estimates will be uncoupled if the two-dimensional illumination function is separable into the product of one-dimensional functions [21:44]. Since, the illumination functions used in the two-dimensional development are Gaussian, the estimates are assumed uncorrelated.

To move the boresight in both azimuth and elevation, the relationship between the two error equations (equations 91 and 93) must be determined. The covariance between azimuthal and elevation errors after lengthy calculations is:

$$\begin{aligned} \text{cov}(e_e, e_a) &= E\{[e_e - E(e_e)][e_a - E(e_a)]\} \\ &= N_0 BW^2/2[\hat{\theta}_{ml}(1 + V_a^2) + \hat{\theta}_{ml}\hat{\phi}(1 + V_e^2) - \hat{\theta}_{ml}\hat{\phi}_{ml}V_s^2] \end{aligned} \tag{95}$$

Evaluating equation 95 at the average estimate values for $\hat{\theta}_{ml}, \hat{\phi}_{ml}$:

$$\text{cov}(e_e, e_a) = N_0 BW^2 \theta\phi/2[V_a^2 + V_e^2 + V_s^2] \tag{95A}$$

Examination of equation 95A shows that in the no-noise environment the covariance will be small for small angular displacements θ, ϕ. Since target tracking implies that both θ and ϕ are less than one-half the beamwidth (1.5 degrees), the coupling coefficient between azimuth and elevation error is assumed negligible. Neglecting the coupling, allows the error voltages to be used to drive two seperate anterna servo loops, similar to the one-dimensional servo loop of Chapter 2, to maintain track-axis alignment with the target.

Discriminators

It is necessary to express the two error equations (equations 91 and 93) as a function of the difference between the estimated and actual target locations. Functions of the diffference allow projected boresight movement along the two orthogonal coordinate axes proportional to the computed difference. With additive noise, the error equations are random quantities as R_1, R_2, and R_3 are independent Gaussian distributed random variables. Assuming no-noise, the observations will be equal to their respective mean values. Using the mean values of the observations R_1, R_2, and R_3 and expressing the elevation difference as d0, the elevation error (equation 91) can be expressed as:

$$e_e = - A\sqrt{E}BW d\theta - A\sqrt{E}\phi^2 d\theta/BW$$

where:

$$d\theta = \theta - \hat{\theta}$$

Solving for $d\theta$ and approximating the result:

$$d\theta = - e_e/Asqrt(E)BW \quad (96)$$

where:

$$(\theta/BW)^2 << 1$$

A similar development for azimuthal error yields an approximate expression for $d\theta$:

$$d\theta = - e_a/Asqrt(E)BW \quad (97)$$

where:

$$(\theta/BW)^2 << 1$$

Examination of equations 96 and 97 indicates that the mapping from estimated error to step function proportional to required boresight movement is linear. Equations 96 an 97 will then be used to actuate seperate elevation and azimuthal type I, improved, servo-control systems to steer the beam-axis on target.

Ziv-Zakai Bound (ZZB)

The Ziv-Zakai two-dimensional bound is derived by comparing the estimation problem with the optimal detection problem. The first development for the Ziv-Zakai bound is to determine the error probability associated with estimating a two-dimensional target position. The next development is to compute the probability of error associated with detection of all possible combinations of the two angular dimensions. This is an M-ary detection problem where M equals 2 squared, or 4. Finally, an inequality must be established between the estimation and detection problems that accounts for all possible target positions in the a priori interval.

Estimation Error Probability

Consider an estimation technique for the two-dimensional angle off boresight S when it is known that the angle is either at position S_1 or position S_2. Figure 45 shows the described situation where the two-dimensional position vector S has azimuthal (ϕ) and elevation (θ) components.

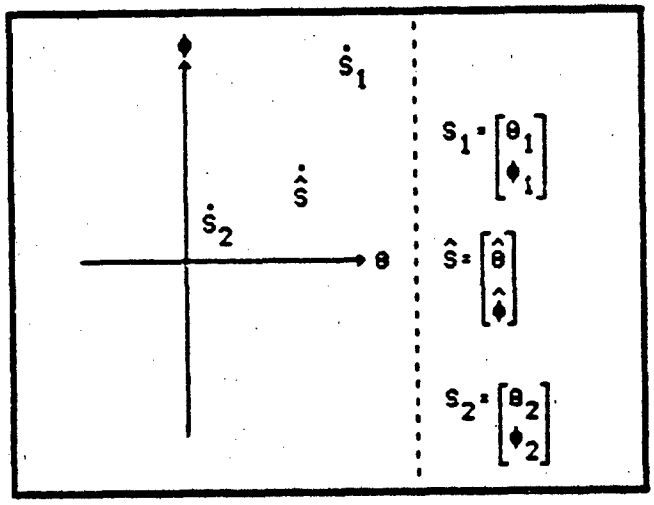

Figure 45. Estimation Geometry

The Ziv-Zakai approach is to compare S with an average value of S, S_{avg}. The decision space for the two possible positions and their average is shown in Figure 46.

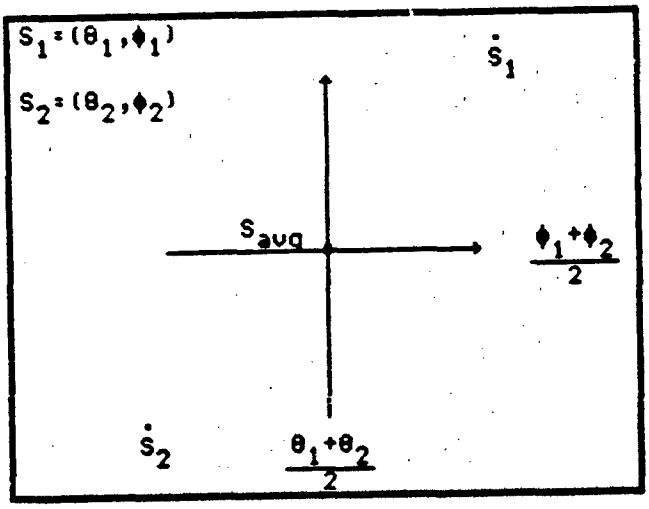

Figure 46. Estimation Decision Space

From the law of total probability:

$$P_e(\text{total}) = P(S_1)P(S < S_{avg}/S_1) + P(S_2)P(S > S_{avg}/S_2)$$

$$= P(S_1)[P((\hat{\theta} < \theta_1/2 + \theta_2/2 \text{ or } \hat{\phi} < \phi_1/2 + \phi_2/2)/S_1) -$$

$$P((\hat{\theta} < \theta_1/2 + \theta_2/2 \text{ and } \hat{\phi} < \phi_1/2 + \phi_2/2)/S_1)] +$$

$$P(S_2)[P((\hat{\theta} > \theta_1/2 + \theta_2/2 \text{ or } \hat{\phi} > \phi_1/2 + \phi_2/2)/S_2) -$$

$$P((\hat{\theta} > \theta_1/2 + \theta_2/2 \text{ and } \hat{\phi} < \phi_1/2 + \phi_2/2)/S_2)] \qquad (98)$$

where:

the first $P(S_1)$ term is left or below the averages

the second $P(S_1)$ term is left and below the averages and accounted for twice, therefore must be subtracted out

similar arguments hold for the $P(S_2)$ terms

Assuming equal a priori probabilities and rearranging terms:

$$P_e = 0.5[P\{(\hat{\theta} < \theta_1/2 + \theta_2/2 \text{ or } \hat{\beta} < \beta_1/2 + \beta_2/2)/S_1\} +$$
$$P\{(\hat{\theta} > \theta_1/2 + \theta_2/2 \text{ or } \hat{\beta} > \beta_1/2 + \beta_2/2)/S_2\}] -$$
$$0.5[P\{(\hat{\theta} < \theta_1/2 + \theta_2/2 \text{ and } \hat{\beta} < \beta_1/2 + \beta_2/2)/S_1\} +$$
$$P\{(\hat{\theta} > \theta_1/2 + \theta_2/2 \text{ and } \hat{\beta} > \beta_1/2 + \beta_2/2)/S_2\}] \qquad (98A)$$

Consider the or terms of equation 98A and let:

$$d\theta = \theta_2 - \theta_1; \quad \theta_2 = d\theta + \theta_1; \quad \theta_1 = \theta_2 - d\theta$$
$$d\beta = \beta_2 - \beta_1; \quad \beta_2 = d\beta + \beta_1; \quad \beta_1 = \beta_2 - d\beta$$

$$P\{(\hat{\theta} < \theta_1/2 + \theta_2/2 \text{ or } \hat{\beta} < \beta_1/2 + \beta_2/2)/S_1\} =$$
$$P\{(\hat{\theta} - \theta_1 < d\theta/2 \text{ or } \hat{\beta} - \beta_1 < d\beta/2)/S_1\} \qquad (98B)$$
$$P\{(\hat{\theta} > \theta_1/2 + \theta_2/2 \text{ or } \hat{\beta} > \beta_1/2 + \beta_2/2)/S_2\} =$$
$$P\{(\hat{\theta} - \theta_2 > -d\theta/2 \text{ or } \hat{\beta} - \beta_2 > -d\beta/2)/S_2\} \qquad (98C)$$

Imposing symmetric target positions let $S_1 = -S_2$ and fix $d\theta$, $d\beta$ at the maximum usable angle off boresight (ie $d\theta = d\beta = -BW$). Equations 98B and 98C become:

$$P\{(\hat{\theta} - \theta_1 < d\theta/2 \text{ or } \hat{\beta} - \beta_1 < d\beta/2)/S_1\} =$$
$$P\{|\hat{\theta} - \theta_1| \geq BW/2/\theta_1 \text{ or } |\hat{\beta} - \beta_1| \geq BW/2/\beta_1\} \qquad (98D)$$
$$P\{(\hat{\theta} - \theta_2 > -d\theta/2 \text{ or } \hat{\beta} - \beta_2 > -d\beta/2)/S_2\} =$$
$$P\{|\hat{\theta} - \theta_1| \geq BW/2/\theta_1 \text{ or } |\hat{\beta} - \beta_1| \geq BW/2/\beta_1\} \qquad (98E)$$

Substitution of $\theta_1 = \theta$ and $\beta_1 = \beta$ into equations 98D and 98E and scaling by 0.5, the terms can be added together. Equation 98F is the result, and represents the or term contribution to the equation 98A estimation probability of error. The or terms can now be expressed as:

$$P\{|\hat{\theta} - \theta| \geq BW/2 \text{ or } |\hat{\phi} - \phi| \geq BW/2\} \tag{98F}$$

To proceed, equation 98F must be transformed into a second moment. Reference 22 provides a two-dimensional generalization of Tchebychev's inequality [22:18]:

$$P\{|\hat{\theta} - E\{\hat{\theta}\}| \geq \lambda \text{sqrt}(\text{var}\{\hat{\theta}\}) \text{ or } |\hat{\phi} - E\{\hat{\phi}\}| \geq \lambda \text{sqrt}(\text{var}\{\hat{\phi}\})\} \leq$$
$$[1 + \text{sqrt}(1 - p^2)]/\lambda^2 \tag{99}$$

where:

p is the correlation coefficient between $\hat{\theta}$ and $\hat{\phi}$

Following a partial development in reference 22, and applying a Biename' inequality of the nature of equation 38:

$$P\{|\hat{\theta} - \theta| \geq BW/2 \text{ or } |\hat{\phi} - \phi| \geq BW/2\} \leq$$
$$E\{\max[(\hat{\theta} - \theta)^2, (\hat{\phi} - \phi)^2]\}[1 + \text{sqrt}(1 - p^2)]/(BW/2)^2 \tag{99A}$$

Assuming $\hat{\theta}$ and $\hat{\phi}$ are uncorrelated:

$$P\{|\hat{\theta} - \theta| \geq BW/2 \text{ or } |\hat{\phi} - \phi| \geq BW/2\} \leq 8\hat{e}^2(S)/BW^2 \tag{99B}$$

where:

$\hat{e}^2(S) = E\{\max[(\hat{\theta} - \theta)^2, (\hat{\phi} - \phi)^2]\}$

= two-dimensional mean-square estimation error

Now, consider the and terms of equation 98A:

$$P\{(\hat{\theta} < \theta_1/2 + \theta_2/2 \text{ and } \hat{\phi} < \phi_1/2 + \phi_2/2)/S_1\} =$$
$$P\{|\hat{\theta} - \theta| \geq BW/2/\theta \text{ and } |\hat{\phi} - \phi| \geq BW/2/\phi\} \tag{100}$$
$$P\{(\hat{\theta} > \theta_1/2 + \theta_2/2 \text{ and } \hat{\phi} > \phi_1/2 + \phi_2/2)/S_2\} =$$
$$P\{|\hat{\theta} - \theta| \geq BW/2/\theta \text{ and } |\hat{\phi} - \phi| \geq BW/2/\phi\} \tag{100A}$$

Scaling by -0.5 and adding equations 100 and 100A results in equation 100B, which represents the and term contribution to the equation 98A estimation probability of error. The and terms can now be expressed as:

$$-P\{|\hat{\theta} - \theta| \geq BW/2 \text{ and } |\hat{\phi} - \phi| \geq BW/2\} \quad (100B)$$

Assuming $\hat{\theta}$ and $\hat{\phi}$ are independent, equation 100B becomes:

$$-[P\{|\hat{\theta} - \theta| \geq BW/2\}P\{|\hat{\phi} - \phi| \geq BW/2\}] \quad (100C)$$

From the one-dimensional development, equation 100C becomes the product of scaled one-dimensional error probabilities:

$$-[\hat{e}^2(\theta)/(BW/2)^2\hat{e}^2(\phi)/(BW/2)^ \quad (100D)$$

where:

$\hat{e}^2(\theta) = E\{|\hat{\theta} - \theta|^2\}$ = mean-square el error

$\hat{e}^2(\phi) = E\{|\hat{\phi} - \phi|^2\}$ = mean-square az error

Combining the results from the or terms (equation 99B) with the results obtained from the and terms (equation 100D), the expression for estimation probability of error (equation 98A) becomes:

$$P_e(\text{est}) = [\hat{e}^2(S) - 2\hat{e}^2(\theta)\hat{e}^2(\phi)/BW^2]8/BW^2 \quad (101)$$

Detection error probability

The detection error probability is the error probability for deciding whether a target is at S_1 or S_2 when it is known to be at one of these two positions with equal probability. The decision regions for the two-dimensional case are shown in Figure 47.

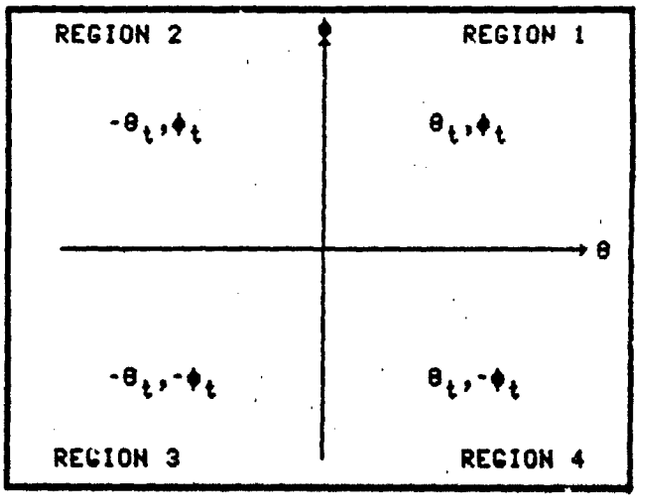

Figure 47. Detection Decision Regions

The four detection hypotheses are:

$$H_1: \quad R_1 = A\text{sqrt}(E)V_s(+\theta_t, +\phi_t) + N_1$$
$$R_2 = A\text{sqrt}(E)V_e(+\theta_t, +\phi_t) + N_2$$
$$R_3 = A\text{sqrt}(E)V_a(+\theta_t, +\phi_t) + N_3$$

$$H_2: \quad R_1 = A\text{sqrt}(E)V_s(-\theta_t, +\phi_t) + N_1$$
$$R_2 = A\text{sqrt}(E)V_e(-\theta_t, +\phi_t) + N_2$$
$$R_3 = A\text{sqrt}(E)V_a(-\theta_t, +\phi_t) + N_3$$

$$H_3: \quad R_1 = A\text{sqrt}(E)V_s(-\theta_t, -\phi_t) + N_1$$
$$R_2 = A\text{sqrt}(E)V_e(-\theta_t, -\phi_t) + N_2$$
$$R_3 = A\text{sqrt}(E)V_a(-\theta_t, -\phi_t) + N_3$$

$$H_4: \quad R_1 = A\text{sqrt}(E)V_s(+\theta_t, -\phi_t) + N_1$$
$$R_2 = A\text{sqrt}(E)V_e(+\theta_t, -\phi_t) + N_2$$
$$R_3 = A\text{sqrt}(E)V_a(+\theta_t, -\phi_t) + N_3$$

Due to the symmetry of the four decision regions of Figure 47, and with equal a priori probabilities, the detection error will be equivalent to the error in a single region. Using hypothesis H_1:

$$P_e\{det\} = P\{l_{12} < 0/H_1\}P\{l_{12}\} + P\{l_{13} < 0/H_1\}P\{l_{13}\} + P\{l_{14} < 0/H_1\}P\{l_{14}\} \quad (102)$$

where:

l_{ij} = natural logarithm of the likelihood ratio

$= \ln[p(r/H_i)/p(r/H_j)]$

Consider the general log-likelihood ratio (l_{ij}):

$$l_{ij} = -N_0^{-1}[(r_1 - E\{R_1/H_i\})^2 + (r_2 - E\{R_2/H_i\})^2 + (r_3 - E\{R_3/H_i\})^2 - (r_1 - E\{R_1/H_j\})^2 - (r_2 - E\{R_2/H_j\})^2 - (r_3 - E\{R_3/H_j\})^2]$$

$$= -[2Asqrt(E)/BWN_0][r_2(\theta_j - \theta_i) + r_3(\phi_j - \phi_i) + 0.5Asqrt(E)/BW(\theta_i^2 - \theta_j^2 + \phi_i^2 - \phi_j^2)]$$

where:

$E\{R_1\}$ = equation 87, and is independent of θ, ϕ

$E\{R_2\}$ = equation 87A, and is independent of ϕ

$E\{R_3\}$ = equation 87B, and is independent of θ

An equivilant decision rule considers l'_{ij} where:

$$l'_{ij} = l_{ij}(BWN_0)/(2Asqrt(E))$$

The general conditional mean of l'_{ij} follows as:

$$E\{l'_{ij}/H_1\} = Asqrt(E)/BW[\theta_t(\theta_i - \theta_j) + 0.5(\theta_j^2 - \theta_i^2 + \phi_j^2 - \phi_i^2) + \phi_t(\phi_i - \phi_j)] \quad (103)$$

A similar analysis on the general conditional variance of l'_{ij} yields:

$$\text{var}\{l'_{ij}/H_1\} = 0.5N_0[(\theta_i - \theta_j)^2 + (\phi_i - \phi_j)^2] \quad (104)$$

To uniquely determine the density functions of the various combinations of l'_{ij}, the statistics must be derived from the general formulas given. With Gaussian statistics for each of the joint conditional densities, l'_{ij} is the sum of independent Gaussian random variables, therefore Gaussian.

Consider the combination i = 1, j = 2; $i = (+\theta_t, +\phi_t)$, $j = (-\theta_t, +\phi_t)$:

$$E\{l'_{12}/H_1\} = 2A\text{sqrt}(E)\theta_t^2/BW; \quad \text{var}\{l'_{12}/H_1\} = 2N_0\theta_t^2 \quad (105)$$

$$p(l'_{12}/H_1) = [k_{12}(\text{pi})]^{-.5}\exp\{-(k_{12})^{-1}(l'_{12} - 2A\text{sqrt}(E)\theta_t^2/BW)^2\}$$

where

$$k_{12} = 4N_0\theta_t^2$$

Consider the combination i = 1, j = 3; $i = (+\theta_t, +\phi_t)$, $j = (-\theta_t, -\phi_t)$:

$$E\{l'_{13}/H_1\} = 2A\text{sqrt}(E)(\theta_t^2+\phi_t^2)/BW; \quad \text{var}\{l'_{13}/H_1\} = 2N_0(\theta_t^2+\phi_t^2) \quad (106)$$

$$p(l'_{13}/H_1) = [k_{13}(\text{pi})]^{-.5}\exp\{-(k_{13})^{-1}(l'_{13} - 2A\text{sqrt}(E)(\theta_t^2 + \phi_t^2)/BW)^2\}$$

where

$$k_{13} = 4N_0(\theta_t^2 + \phi_t^2)$$

Consider the combination i = 1, j = 4; $i = (+\theta_t, +\phi_t)$, $j = (+\theta_t, -\phi_t)$:

$$E\{l'_{14}/H_1\} = 2A\text{sqrt}(E)\phi_t^2/BW; \quad \text{var}\{l'_{14}/H_1\} = 2N_0\phi_t^2 \quad (107)$$

$$p(l'_{14}/H_1) = [k_{14}(\text{pi})]^{-.5}\exp\{-(k_{14})^{-1}(l'_{14} - 2A\text{sqrt}(E)\phi_t^2/BW)^2\}$$

where

$$k_{14} = 4N_0\phi_t^2$$

Using the appropriate change of variables in each of the three density

functions described by equations 105, 106, and 107, the detection error (equation 102) can now be expressed as:

$$P_e\{det\} = P\{l'_{12}\}Q[E\{l'_{12}/H_1\}/sqrt(var\{l'_{12}/H_1\})] + P\{l'_{13}\}Q[E\{l'_{13}/H_1\}/sqrt(var\{l'_{13}/H_1\})] + P\{l'_{14}\}Q[E\{l'_{14}/H_1\}/sqrt(var\{l'_{14}/H_1\})] \quad (108)$$

where:

$$Q[a] = sqrt(2(pi))^{-1}\int_a^{\infty} exp[-x^2/2]dx$$

statistics of l'_{12} are given in equation 105

statistics of l'_{13} are given in equation 106

statistics of l'_{14} are given in equation 107

Substituting the appropriate statistics into equation 108 and simplifying:

$$P_e\{det\} = 0.33[Q(A\theta_t sqrt(2SNR)/BW) + Q(A\phi_t sqrt(2SNR)/BW) + Q(Asqrt(\theta_t^2 + \phi_t^2)sqrt(2SNR)/BW)] \quad (108A)$$

where:

$$SNR = E/N_0$$

$$P\{l'_{12}\} = P\{l'_{13}\} = P\{l'_{14}\} = 1/3$$

Equation 108A describes the P_e of the best procedure for deciding whether a target is at S_1 or $-S_1$, when it is known to be at one of these two locations with equal probabality. To compare with the results obtained in one-dimensional tracking consider $\theta_t = \phi_t = 0$:

$$P_e\{det(0,0)\} = .333[Q(0) + Q(0) + Q(0)] = 0.5$$

This result is the same as obtained in the one-dimensional case where the target seperation angle was zero. Here, target seperation in two

orthogonal planes is zero and once again the detection error indicates that the targets will be difficult to distinguish.

Detection/Estimation Inequality

Following the Ziv-Zakai method, a performance bound can be derived by comparing the estimation problem with the optimal detection problem. The estimation decision rule error will be lower bounded by the error associated with an optimal detection scheme. Establishing an estimation/detection inequality:

$$P_e(\text{det}) \leq P_e(\text{est}) \tag{109}$$

Using the estimation probability of error as described by equation 101 and solving equation 109 for the two-dimensional mean-square estimation error:

$$\hat{e}^2(S) \geq BW^2 P_e\{S,-S\}/8 + 2\hat{e}^2(\theta)\hat{e}^2(\phi)/BW^2 \tag{109A}$$

where:

$P_e\{S,-S\}$ is described by equation 108A

$\hat{e}^2(a)$ refers to the mean-square error for parameter a and given by the expression in equation 40C

Equation 109A is a lower bound to the mean-square estimation error of any pair of values of the parameter S which are BW units apart. Extension of equation 109A to consider all possible positions in the a priori interval of the parameter S, and using the worst case error of all possible a priori values results in:

$$\hat{e}^2(S) \geq \max_{\substack{0 \leq \theta_t \leq \theta_{max} \\ 0 \leq \phi_t \leq \phi_{max}}} [S^2 P_e(S,-S)/8 + 2\hat{e}^2(\theta)_t \hat{e}^2(\phi)_t / S^2] \qquad (109B)$$

where:

$$S = \begin{bmatrix} \theta_t \\ \phi_t \end{bmatrix}$$

Observe in equation 109B that the lower bound does not exist at S = 0. Thus a restriction on the bound as described in references 9 and 12, and previously used in equation 40C of this thesis, must be imposed. Further, because of the symmetry invoked, the bound only considers positive values of azimuth and elevation. Positive parameter values allow the substitution of the magnitude squared of the parameter for the square of the parameter. In vector notation:

$$S^2 \text{ (pos S)} = |S|^2 = SS^T = \begin{bmatrix} \theta_t \\ \phi_t \end{bmatrix} [\theta_t \quad \phi_t]$$

$$= \theta_t^2 + \phi_t^2$$

Redefining the bound of equation 109B:

$$\hat{e}^2(S) \geq \max_{\substack{0 \leq \theta_t \leq \theta_p \\ 0 \leq \phi_t \leq \phi_p}} [\sin(|S|)^2 P_e(S,-S)/8 + 2\hat{e}^2(\theta)_t \hat{e}^2(\phi)_t / \sin(|S|)^2] \qquad (109C)$$

where:

$$\sin(\theta_p) = 0.5[\sin(\theta_{max}) + \sin(\theta)]$$

$$\sin(\phi_p) = 0.5[\sin(\phi_{max}) + \sin(\phi)]$$

Equation 109C will be the form of the Ziv-Zakai bound used to predict the mean-square estimation error for maximum-likelihood estimates of elevation and azimuthal angles.

Radar Simulation

To compute the mean-square tracking error and associated performance bounds of the two-dimensional amplitude-comparison monopulse radar, a simulation program was designed for the radar model as developed in this appendix. An annotated listing of the simulation program is provided in Appendix G. The program is a straightforward extension of the one-dimensional simulation provided in Appendix B. The major modifications and significant features are:

1) Three independent observations are required.
2) The Cramer-Rao bound is not computed.
3) Provisions for moving targets are removed.
4) Equation 109C (ZZB) is never evaluated at $\theta = \beta =$ precisely zero. The slight offset from zero allows computation of the second term of the described equation.
5) A storage matrix (16X16) is established for the evaluation of equation 109C (ZZB) for the various segmented angle intervals in azimuth and elevation. The stored values are then sorted by row and column, and the maximum is picked off.

Performance Verification

Before comparing the results with the one-dimensional tracking results, a few comment about the simulation is in order. For each computed bound of the simulation, the Q function is evaluated by integrating the tails of a standard Normal density in an external call to the IMSL subroutine MDNOR. One run calls the MDNOR subroutine

approximately thirteen hundred times. Each SNR level conducts fifteen runs, therefore approximately nineteen thousand integrations. For lower SNR levels, the area integrated increases, and the time consumed for each run is quite lengthy. Low SNR runs were not possible because of time limits on the computer system used (CYBER).

Figure 48 is a stationary target composite plot for the two-dimensional target mean-square tracking error (MSE), and the Ziv-Zakai bound (ZZB) for SNR levels of 30, 25, 15, and 7 dB. Comparision of Figure 48 with the previous stationary one-dimensional target tracking results demonstrate that:

1) The MSE is much larger for the two-dimensional tracking case [reference approximately 3X10E-5 for one-dimension at 15dB SNR (see Figure 30) and approximately 20X10E-5 for two-dimensions at 15dB SNR (see Figure 48)].
2) At 7dB SNR, the two-dimensional MSE grows unbounded. Breaklock is assumed to occur at this point. This was not encountered in the one-dimensional tracker.
3) For all SNR levels considered, the two-dimensional ZZB lower bounds the MSE curves. Although not as tight as in the one-dimensional tracker, the two-dimensional ZZB is an adequate performance indicator, and merits further investigation.

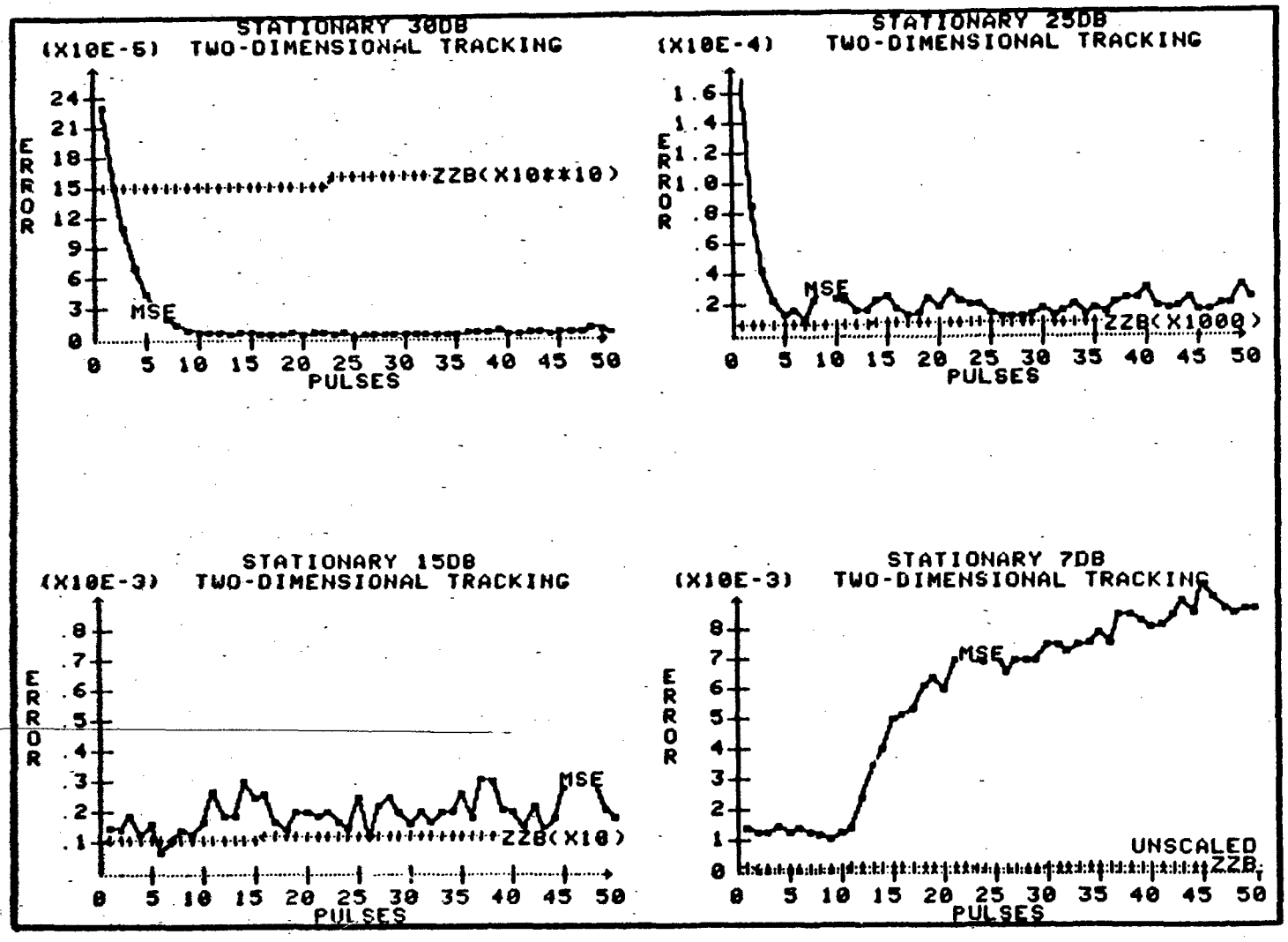

Figure 48. Two-Dimensional Target Tracking Error vs SNR Level

Appendix B

Program Listing

```
*************************************************************************
*     PROGRAM:    SIMULATION OF AMPLITUDE-COMPARISON MONOPULSE RADAR     *
*     PURPOSE:    COMPUTE THE TRACKING ERROR AND PERFORMANCE BOUNDS      *
*     INPUT:      DESIRED SNR LEVEL(dB), TARGET MOVING OR STATIONARY     *
*     OUTPUT:     3(1 BY 50) VECTORS CONTAINING AVG ERROR, CRB, ZZB      *
*     COMPILER:   FORTRAN, VERSION 5                                     *
*     EXT CALLS:  IMSL ROUTINES GGNML, MDNOR                             *
*************************************************************************
      PROGRAM ACMON                                                    *                *
                                                                       * DECLARE VARIABLES *
      DIMENSION AVG(50),AVG1(50),AVG2(50),PRE(16),R(50),B(50)          *                *
      DIMENSION SNRV(15),SER(15,50),SERCR(15,50),SERZZ(15,50)          *                *
      REAL NPWR,N1,N2,NAMP,NCALC                                       *   INTEGER TO REAL *
      INTEGER PULSE,ANGLE                                              *   REAL TO INTEGER *
      DOUBLE PRECISION DSEED1,DSEED2                                   *   DOUBLE PRECISION SEEDS *
                                                                       * INPUT STATEMENTS  *
      DATA SNRV/35.,30.,25.,20.,15.,10.,7.,5.,3.,1.,                   *                *
     *-2.,-5.,-7.,-10.,-20./                                           *   POSSIBLE SNR VALUES (dB)*
 5    CONTINUE                                                         *                *
      WRITE(*,1000)                                                    *                *
      READ*,IMS                                                        *   TARGET MOVING OR QUIT *
      IF(IMS.EQ.0)GO TO 5000                                           *                *
      WRITE(*,1010)                                                    *                *
      WRITE(*,1020)                                                    *                *
      READ*,I1,I2                                                      *   START/STOP SNR VALUES *
                                                                       * RADAR PARAMETERS  *
      DATA DSEED1,DSEED2/123457.0D0,325017.0D0/                        *                *
      DATA DPRAD,R0,VEL,PW,BW/57.3,4900.,257.4,1.E-6,3.0/              *                *
      DATA PRI,SIG,OMEGA,AMP,SPWR/10.E-3,636.,636.,5.0,1.E-3/          *                *
      EGY=PW*AMP**2                                                    *   ENERGY       *
      NR=50                                                            *   50 GAUSSIAN SAMPLES *
      NB=50                                                            *   50 GAUSSIAN SAMPLES *
      DO 500 IP=I1,I2                                                  *   SNR DO LOOP  *
      SNR=SNRV(IP)                                                     *                *
      SNRNEW=10.**(SNR/10.)                                            *   CONVERT TO NUMERIC *
      ROLD=R0                                                          *   INITIAL RANGE *
      DO 200 J=1,15                                                    *   J=RUN NUMBER *
      CALL GGNML(DSEED1,NR,R)                                          *   NORMAL DEVIATE GENERATOR*
      CALL GGNML(DSEED2,NB,B)                                          *   NORMAL DEVIATE GENERATOR*
                                                                       * PLACE TARGET IN BEAMWIDTH *
      BSITE=0.0                                                        *   STATIONARY   *
      TRGT=0.00                                                        *   STATIONARY   *
      IF(IMS.EQ.1)BSITE=1.5                                            *   MOVING       *
      IF(IMS.EQ.1)TRGT=0.0                                             *   MOVING       *
                                                                       * TRACKING CALCULATIONS *
      THETA=TRGT-BSITE                                                 *   TARGET ANGLE OF ARRIVAL *
      I=0                                                              *                *
      DO 100 PULSE=1,50                                                *   50 PULSES    *
      PULSEC=FLOAT(PULSE)                                              *   CONVERT TO NUMERIC *
```

```
      RNEW=SQRT(R0**2+(VEL*PRI*PULSEC)**2)                    *   COMPUTE CHANGING RANGE  *
      IF(IMS.EQ.1.)SNRNEW=((ROLD/RNEW)**4)*SNRNEW             *   COMPUTE CHANGING SNR    *
      NPWR=SPWR/SNRNEW                                        *   NOISE POWER             *
      NAMP=SQRT(NPWR)                                         *   NOISE AMPLITUDE         *
      I=I+1                                                   *                           *
      N1=R(I)*NAMP                                            *   WGN SAMPLE OBS 1        *
      N2=B(I)*NAMP                                            *   WGN SAMPLE OBS 2        *
      VD=SIN(1.88*THETA/BW)                                   *   ANTENNA DIFF WEIGHTING  *
      VS=(SQRT(2.)/2.)*(1.+COS(2.094*THETA/BW))               *   ANTENNA SUM WEIGHTING   *
      R1=AMP*SQRT(EGY)*VS+N1                                  *   SUM CHANNEL OBSERVATION *
      R2=AMP*SQRT(EGY)*VD+N2                                  *   DIFF CHANNEL OBSERVATION*
      ERR=R1*VD-R2*VS                                         *   ERROR SIGNAL            *
      IF(ERR/(SQRT(2.*EGY)*AMP).GE.0.99) ERR=0.99*SQRT(2.*EGY)*AMP  *   LIMIT POSITIVE ERROR  *
      IF(ERR/(SQRT(2.*EGY)*AMP).LE.-0.99) ERR=-0.99*SQRT(2.*EGY)*AMP *   LIMIT NEGATIVE ERROR  *
      DTHETA=-(BW/1.88)*ASIN(ERR/(SQRT(2.*EGY)*AMP))          *   STEP PROPORTIONAL TO ERR*
      SERVO=1.0-EXP(-SIG*PRI)*(COS(OMEGA*PRI)-SIN(OMEGA*PRI)) *   SERVO RESPONSE          *
      IF(VD*ERR.GT.0.0)DTHETA=-DTHETA                         *   TRACK PROPERLY          *
      BSITE=BSITE+SERVO*DTHETA                                *   MOVE BORESIGHT          *
      IF(IMS.EQ.1)TRGT=ATAN((VEL*PULSEC*PRI)/R0)*DPRAD        *   CHANGING TARGET POSITION*
      THETA=TRGT-BSITE                                        *   NEW ANGLE OF ARRIVAL    *
      BW=BW/DPRAD                                             *                           *
      THETA=THETA/DPRAD                                       *                           *
                                                              * TRACKING ERROR            *
      SER(J,PULSE)=THETA**2                                   *   SQUARED ERROR           *
                                                              * COMPUTE CRAMER-RAO BOUND   *
      DENOM=(2.*SNRNEW*AMP**2)*(2.845+1.77*COS(3.76*THETA/BW) *                           *
     *-1.895*COS(4.188*THETA/BW))                             *   DENOMINATOR             *
      CRB=(BW**2)/DENOM                                       *   BOUND                   *
      SERCR(J,PULSE)=CRB                                      *                           *
                                                              * COMPUTE ZIV-ZAKAI BOUND   *
      THETP=(ASIN((SIN(BW/2)+SIN(ABS(THETA)))*0.5))*DPRAD     *   ARITHMETIC MEAN ANGLE   *
      BW=BW*DPRAD                                             *                           *
      THETA=THETA*DPRAD                                       *                           *
      DATA PRE/16*0.0/                                        *   INITIALIZE              *
      ANGLE=1.+ABS(INT(10.*THETP))                            *   RANGE X10 FOR DO LOOP   *
      DO 90 N=1,ANGLE                                         *   CHECK FROM 0-THETP      *
      NCALC=(FLOAT(ANGLE)/100.)*N                             *                           *
      W=5./2+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)   *                           *
     *-COS(3.76*NCALC/BW)                                     *   ANTENNA WEIGHTING       *
      RHO=(0.5+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW) *                           *
     *+COS(3.76*NCALC/BW))/W                                  *   CORRELATION COEFFICIENT *
      AX=(SQRT((1.-RHO)*W*SNRNEW/2.))*AMP                     *   INPUT PARAMETER AX      *
      CALL MDNOR(AX,PX)                                       *   COMPUTE AREA -INF TO AX *
      QX=1.-PX                                                *   Q FUNCTION (Pe)         *
      NCALC=NCALC/DPRAD                                       *                           *
      PRE(N)=QX*(SIN(1.*NCALC))**2                            *   BOUND                   *
90    CONTINUE                                                *                           *
      DO 94 JK=1,15                                           *   SORT ROUTINE PRE()      *
      JK1=JK+1                                                *                           *
      DO 92 KJ=JK1,16                                         *                           *
      IF(PRE(JK).LE.PRE(KJ))GO TO 92                          *                           *
      STORE=PRE(JK)                                           *                           *
      PRE(JK)=PRE(KJ)                                         *                           *
```

```
      PRE(KJ)=STORE
   92 CONTINUE
   94 CONTINUE
      ZZB=PRE(16)
      SERZZ(J,PULSE)=ZZB

  100 CONTINUE
  200 CONTINUE
      WRITE(*,1030)SNR
      WRITE(*,1040)

      DO 400 K=1,50
      SUM=0.0
      SUM1=0.0
      SUM2=0.0
      DO 300 L=1,15
      SUM=SUM+SER(L,K)
      SUM1=SUM1+SERCR(L,K)
      SUM2=SUM2+SERZZ(L,K)
  300 CONTINUE
      AVG(K)=SUM/15.
      AVG1(K)=SUM1/15.
      AVG2(K)=SUM2/15.

      WRITE(*,1050)AVG(K),AVG1(K),AVG2(K)
  400 CONTINUE
  500 CONTINUE
      GO TO 5

 1000 FORMAT(1X,'MOVING TGT,STAT TGT, OR QUIT(1:MOVE,2:STAT,0:QUIT)')
 1010 FORMAT(1X,'SNRV/35;30,25,20,15,10,7,5,3,1,-2,-5,-7,-10,-20/',/)
 1020 FORMAT(1X,'ENTER SNRV START/STOP POINTS(IE START=1,STOP=15)')
 1030 FORMAT(/,1X,'SNR LEVEL (DB) = ',E11.5,/)
 1040 FORMAT(14X,'AVGSER',8X,'AVGSERCR',6X,'AVGSERZZ')
 1050 FORMAT(10X,3(3X,E11.5))
 5000 STOP
      END
```

PRE(16) IS MAXIMUM

HEADER
AVERAGE ERRORS OVER 15 RUNS

OUTPUT RESULTS

ONLY WAY OUT IS 0 IN IMS
FORMAT STATEMENTS IN/OUT

Appendix C

Program Listing

```
*************************************************************************
*     PROGRAM:    SIMULATION OF "SLOW" AND "RAPID" AMPLITUDE FADING     *
*     PURPOSE:    COMPUTE THE TRACKING ERROR AND PERFORMANCE BOUNDS     *
*     INPUT:      DESIRED SNR LEVEL(dB), TARGET MOVING OR STATIONARY,   *
*                 TYPE OF FADE CHARACTERISTIC DESIRED                   *
*     OUTPUT:     3(1 BY 50) VECTORS CONTAINING AVG ERROR, CRB, ZZB     *
*     COMPILER:   FORTRAN, VERSION 5                                    *
*     EXT CALLS:  IMSL ROUTINES GGNML, MDNOR, GGWIB                     *
*************************************************************************
      PROGRAM FADE                                                    *
                                                                      *  DECLARE VARIABLES
      DIMENSION AVG(50),AVG1(50),AVG2(50),PRE(16),R(50),B(50),A(50)   *    50 AMPLITUDE SAMPLES
      DIMENSION SNRV(15),SER(15,50),SERCR(15,50),SERZZ(15,50)         *
      REAL NPWR,N1,N2,NAMP,NCALC                                      *
      INTEGER PULSE,ANGLE,SWERLG                                      *    FADE TYPE:  SWERLG
      DOUBLE PRECISION DSEED1,DSEED2                                  *
                                                                      *  INPUT STATEMENTS
      DATA SNRV/35.,30.,25.,20.,15.,10.,7.,5.,3.,1.,                  *
     *-2.,-5.,-7.,-10.,-20./                                          *
 5    CONTINUE                                                        *
      WRITE(*,1000)                                                   *
      READ*,IMS                                                       *
      IF(IMS.EQ.0)GO TO 5000                                          *
      WRITE(*,1010)                                                   *
      WRITE(*,1020)                                                   *
      READ*,I1,I2                                                     *
      WRITE(*,1025)                                                   *
      READ*,SWERLG                                                    *    INPUT FADE TYPE
                                                                      *  RADAR PARAMETERS
      DATA DSEED1,DSEED2/123457.000,325817.000/                       *
      DATA DPRAD,RO,VEL,PW,BW/57.3,4900.,257.4,1.E-6,3.0/             *
      DATA PRI,SIG,OMEGA,SPWR/10.E-3,636.,636.,1.E-3/                 *
      ALPHA=3.9894                                                    *    RAYLEIGH PARAMETER-STDEV
      S=2                                                             *    SHAPE PARAMETER
      T=SQRT(2.)*ALPHA                                                *    SCALE PARAMETER
      NR=50                                                           *
      NB=50                                                           *
      NA=50                                                           *    50 RAYLEIGH SAMPLES
      IF (SWERLG.EQ.1)NA=15                                           *    15 SAMPLES SLOW FADE
      DO 500 IP=I1,I2                                                 *
      SNR=SNRV(IP)                                                    *
      SNRNEW=10.**(SNR/10.)                                           *
      ROLD=RO                                                         *
      DO 200 J=1,15                                                   *
      CALL GGNML(DSEED1,NR,R)                                         *
      CALL GGNML(DSEED2,NB,B)                                         *
      CALL GGWIB(DSEED1,S,NA,A)                                       *    WEIBULL DEVIATES
```

```
                                                    * PLACE TARGET IN BEAMWIDTH   *
BSITE=0.0                                           *                             *
TRGT=0.00                                           *                             *
IF(IMS.EQ.1)BSITE=1.5                               *                             *
IF(IMS.EQ.1)TRGT=0.0                                *                             *
                                                    * TRACKING CALCULATIONS       *
THETA=TRGT-BSITE                                    *                             *
I=0                                                 *                             *
DO 100 PULSE=1,50                                   *                             *
PULSEC=FLOAT(PULSE)                                 *                             *
RNEW=SQRT(R0**2+(VEL*PRI*PULSEC)**2)                *                             *
IF(IMS.EQ.1.)SNRNEW=((ROLD/RNEW)**4)*SNRNEW         *                             *
NPWR=SPWR/SNRNEW                                    *                             *
NAMP=SQRT(NPWR)                                     *                             *
I=I+1                                               *                             *
N1=R(I)*NAMP                                        *                             *
N2=B(I)*NAMP                                        *                             *
IF(SWERLG.NE.1)AMP=T*A(I)                           *   "RAPID" FADE AMP          *
IF(SWERLG.EQ.1)AMP=T*A(J)                           *   "SLOW" FADE AMP           *
EGY=PW*AMP**2                                       *   EGY CHANGES WITH A        *
VD=SIN(1.88*THETA/BW)                               *                             *
VS=(SQRT(2.)/2.)*(1.+COS(2.094*THETA/BW))           *                             *
R1=AMP*SQRT(EGY)*VS+N1                              *                             *
R2=AMP*SQRT(EGY)*VD+N2                              *                             *
ERR=R1*VD-R2*VS                                     *                             *
IF(ERR/(SQRT(2.*EGY)*AMP).GE.0.99) ERR=0.99*SQRT(2.*EGY)*AMP  *                   *
IF(ERR/(SQRT(2.*EGY)*AMP).LE.-0.99) ERR=-0.99*SQRT(2.*EGY)*AMP *                  *
DTHETA=-(BW/1.88)*ASIN(ERR/(SQRT(2.*EGY)*AMP))      *                             *
SERVO=1.0-EXP(-SIG*PRI)*(COS(OMEGA*PRI)-SIN(OMEGA*PRI))  *                        *
IF(VD*ERR.GT.0.0)DTHETA=-DTHETA                     *                             *
BSITE=BSITE+SERVO*DTHETA                            *                             *
IF(IMS.EQ.1)TRGT=ATAN((VEL*PULSEC*PRI)/R0)*DPRAD    *                             *
THETA=TRGT-BSITE                                    *                             *
BW=BW/DPRAD                                         *                             *
THETA=THETA/DPRAD                                   *                             *
                                                    * TRACKING ERROR              *
SER(J,PULSE)=THETA**2                               *                             *
                                                    * COMPUTE CRAMER-RAO BOUND     *
DENOM=(2.*SNRNEW*AMP**2)*(2.865+1.77*COS(3.76*THETA/BW)  *                        *
*-1.095*COS(4.188*THETA/BW))                        *                             *
CRB=(BW**2)/DENOM                                   *                             *
SERCR(J,PULSE)=CRB                                  *                             *
                                                    * COMPUTE ZIV-ZAKAI BOUND      *
THETP=(ASIN((SIN(BW/2)+SIN(ABS(THETA)))*0.5))*DPRAD *                             *
BW=BW*DPRAD                                         *                             *
THETA=THETA*DPRAD                                   *                             *
DATA PRE/16*0.0/                                    *                             *
ANGLE=1.+ABS(INT(10.*THETP))                        *                             *
DO 90 N=1,ANGLE                                     *                             *
NCALC=(FLOAT(ANGLE)/100.)*N                         *                             *
W=5./2+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)  *                          *
*-COS(3.76*NCALC/BW)                                *                             *
RHO=(0.5+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)  *                        *
```

```
     *+COS(3.76*NCALC/BW)/W
      AX=(SQRT((1.-RHO)*W*SNRNEW/2.))*AMP
      CALL MONOR(AX,PX)
      QX=1.-PX
      NCALC=NCALC/DPRAD
      PRE(N)=QX*(SIN(1.*NCALC))**2
   90 CONTINUE
      DO 94 JK=1,15
      JK1=JK+1
      DO 92 KJ=JK1,16
      IF(PRE(JK).LE.PRE(KJ))GO TO 92
      STORE=PRE(JK)
      PRE(JK)=PRE(KJ)
      PRE(KJ)=STORE
   92 CONTINUE
   94 CONTINUE
      ZZB=PRE(16)
      SERZZ(J,PULSE)=ZZB

  100 CONTINUE
  200 CONTINUE
      WRITE(*,1030)SNR
      WRITE(*,1040)

      DO 400 K=1,50
      SUM=0.0
      SUM1=0.0
      SUM2=0.0
      DO 300 L=1,15
      SUM=SUM+SER(L,K)
      SUM1=SUM1+SERCR(L,K)
      SUM2=SUM2+SERZZ(L,K)
  300 CONTINUE
      AVG(K)=SUM/15.
      AVG1(K)=SUM1/15.
      AVG2(K)=SUM2/15.

      WRITE(*,1050)AVG(K),AVG1(K),AVG2(K)
  400 CONTINUE
  500 CONTINUE
      GO TO 5

 1000 FORMAT(1X,'MOVING TGT,STAT TGT, OR QUIT(1:MOVE,2:STAT,0:QUIT)')
 1010 FORMAT(1X,'SNRV/35,30,25,20,15,10,7,5,3,1,-2,-5,-7,-10,-20/',/)
 1020 FORMAT(1X,'ENTER SNRV START/STOP POINTS(IE START=1,STOP=15)')
 1025 FORMAT(1X,'ENTER DESIRED FADE MODEL (SLOW=1,FAST=2)')
 1030 FORMAT(/,1X,'SNR LEVEL (DB) = ',E11.5,/)
 1040 FORMAT(14X,'AVGSER',8X,'AVGSERCR',6X,'AVGSERZZ')
 1050 FORMAT(10X,3(3X,E11.5))
 5000 STOP
      END
```

AVERAGE ERRORS OVER 15 RUNS

OUTPUT RESULTS

FORMAT STATEMENTS IN/OUT

Appendix D

Program Listing

```
*************************************************************************
*     PROGRAM:     SMOOTHING OF "RAPID" AMPLITUDE FADING OVER N PULSES   *
*     PURPOSE:     COMPUTE THE TRACKING ERROR AND PERFORMANCE BOUNDS     *
*     INPUT:       DESIRED SNR LEVEL(dB), TARGET MOVING OR STATIONARY,   *
*                  NUMBER OF INTEGRATED PULSES DESIRED                   *
*     OUTPUT:      3(1 BY 50) VECTORS CONTAINING AVG ERROR, CRB, ZZB     *
*     COMPILER:    FORTRAN, VERSION 5                                    *
*     EXT CALLS:   IMSL ROUTINES GGNML, MDNOR, GGWIB                     *
*************************************************************************
      PROGRAM FADFIX                                                    *
                                                                        * DECLARE VARIABLES
      DIMENSION AVG(50),AVG1(50),AVG2(50),PRE(16),R(50),B(50),A(50)     *
      DIMENSION SNRV(15),SER(15,50),SERCR(15,50),SERZZ(15,50)           *
      REAL NPWR,N1,N2,NAMP,NCALC                                        *
      INTEGER PULSE,ANGLE                                               *
      DOUBLE PRECISION DSEED1,DSEED2                                    *
      DATA SNRV/35.,30.,25.,20.,15.,10.,7.,5.,3.,1.,                    *
     *-2.,-5.,-7.,-10.,-20./                                            *
 5    CONTINUE                                                          *
      WRITE(*,1000)                                                     *
      READ*,IMS                                                         *
      IF(IMS.EQ.0)GO TO 5000                                            *
      WRITE(*,1010)                                                     *
      WRITE(*,1020)                                                     *
      READ*,I1,I2                                                       *
      WRITE(*,1025)                                                     *
      WRITE(*,1027)                                                     *
      READ*,M                                                           *   M=PULSES INTEGRATED
                                                                        * RADAR PARAMETERS
      DATA DSEED1,DSEED2/123457.0D0,325017.0D0/                         *
      DATA DPRAD,RO,VEL,PW,BW/57.3,4900.,257.4,1.E-6,3.0/               *
      DATA PRI,SIG,OMEGA,SPWR/10.E-3,636.,636.,1.E-3/                   *
      ALPHA=3.9894                                                      *
      S=2                                                               *
      T=SQRT(2.)*ALPHA                                                  *
      NR=50                                                             *
      NB=50                                                             *
      NA=50                                                             *
      DO 500 IP=I1,I2                                                   *
      SNR=SNRV(IP)                                                      *
      SNRNEW=10.**(SNR/10.)                                             *
      ROLD=RO                                                           *
      DO 200 J=1,15                                                     *
      CALL GGNML(DSEED1,NR,R)                                           *
      CALL GGNML(DSEED2,NB,B)                                           *
      CALL GGWIB(DSEED1,S,NA,A)                                         *
                                                                        * PLACE TARGET IN BEAMWIDTH
      BSITE=0.0                                                         *
      TRGT=0.80                                                         *
```

```
      IF(IMS.EQ.1)BSITE=1.5                                          *                          *
      IF(IMS.EQ.1)TRGT=0.0                                           *                          *
                                                                     *  TRACKING CALCULATIONS   *
      THETA=TRGT-BSITE                                               *                          *
      I=0                                                            *                          *
      MQ=INT(50./FLOAT(M))                                           *  MQ=NUMBER OF BSITE MOVES*
      DO 100 PULSE=1,MQ                                              *                          *
      DTSUM=0.0                                                      *  INITIALIZE ERROR SUM    *
      PULSEC=0                                                       *                          *
      DO 150 INTGR=1,M                                               *  SMOOTHING DO LOOP       *
      PULSEC=PULSEC+1                                                *                          *
      RNEW=SQRT(R0**2+(VEL*PRI*PULSEC)**2)                           *                          *
      IF(IMS.EQ.1.)SNRNEW=((ROLD/RNEW)**4)*SNRNEW                    *                          *
      NPWR=SPWR/SNRNEW                                               *                          *
      NAMP=SQRT(NPWR)                                                *                          *
      I=I+1                                                          *                          *
      N1=R(I)*NAMP                                                   *                          *
      N2=B(I)*NAMP                                                   *                          *
      AMP=T*A(I)                                                     *                          *
      EGY=PW*AMP**2                                                  *                          *
      VD=SIN(1.88*THETA/BW)                                          *                          *
      VS=(SQRT(2.)/2.)*(1.+COS(2.094*THETA/BW))                      *                          *
      R1=AMP*SQRT(EGY)*VS+N1                                         *                          *
      R2=AMP*SQRT(EGY)*VD+N2                                         *                          *
      ERR=R1*VD-R2*VS                                                *                          *
      IF(ERR/(SQRT(2.*EGY)*AMP).GE.0.99) ERR=0.99*SQRT(2.*EGY)*AMP   *                          *
      IF(ERR/(SQRT(2.*EGY)*AMP).LE.-0.99) ERR=-0.99*SQRT(2.*EGY)*AMP *                          *
      DTHETA=-(BW/1.88)*ASIN(ERR/(SQRT(2.*EGY)*AMP))                 *                          *
      SERVO=1.0-EXP(-SIG*PRI)*(COS(OMEGA*PRI)-SIN(OMEGA*PRI))        *                          *
      IF(VD*ERR.GT.0.0)DTHETA=-DTHETA                                *                          *
      DTSUM=DTSUM+DTHETA                                             *  ACCUMULATE ERROR SUM    *
  150 CONTINUE                                                       *                          *
      DTHETA=DTSUM/FLOAT(M)                                          *  AVERAGE ERROR           *
      BSITE=BSITE+SERVO*DTHETA                                       *                          *
      IF(IMS.EQ.1)TRGT=ATAN((VEL*PULSEC*PRI)/R0)*DPRAD               *                          *
      THETA=TRGT-BSITE                                               *                          *
      BW=BW/DPRAD                                                    *                          *
      THETA=THETA/DPRAD                                              *                          *
                                                                     *  TRACKING ERROR & BOUNDS *
      SER(J,PULSE)=THETA**2                                          *  TRACKING ERROR          *
      DENOM=(2.*SNRNEW*AMP**2)*(2.865+1.77*COS(3.76*THETA/BW)        *                          *
     *-1.095*COS(4.188*THETA/BW))                                    *                          *
      CRB=(BW**2)/DENOM                                              *  CRAMER-RAO BOUND        *
      SERCR(J,PULSE)=CRB                                             *                          *
      THETP=(ASIN((SIN(BW/2)+SIN(ABS(THETA)))*0.5))*DPRAD            *                          *
      BW=BW*DPRAD                                                    *                          *
      THETA=THETA*DPRAD                                              *                          *
      DATA PRE/16*0.0/                                               *                          *
      ANGLE=1.+ABS(INT(10.*THETP))                                   *                          *
      DO 90 N=1,ANGLE                                                *                          *
      NCALC=(FLOAT(ANGLE)/100.)*N                                    *                          *
      W=5./2+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)          *                          *
     *-COS(3.76*NCALC/BW)                                            *                          *
```

```
      RHO=(0.5+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)
     *+COS(3.76*NCALC/BW))/4
      AX=(SQRT((1.-RHO)*W*SNRNEW/2.))*AMP
      CALL MONOR(AX,PX)
      QX=1.-PX
      NCALC=NCALC/DPRAD
      PRE(N)=QX*(SIN(1.*NCALC))**2
   90 CONTINUE
      DO 94 JK=1,15
      JK1=JK+1
      DO 92 KJ=JK1,16
      IF(PRE(JK).LE.PRE(KJ))GO TO 92
      STORE=PRE(JK)
      PRE(JK)=PRE(KJ)
      PRE(KJ)=STORE
   92 CONTINUE
   94 CONTINUE
      ZZB=PRE(16)
      SERZZ(J,PULSE)=ZZB
  100 CONTINUE
  200 CONTINUE
      WRITE(*,1030)SNR
      WRITE(*,1040)

      DO 400 K=1,50
      SUM=0.0
      SUM1=0.0
      SUM2=0.0
      DO 300 L=1,15
      SUM=SUM+SER(L,K)
      SUM1=SUM1+SERCR(L,K)
      SUM2=SUM2+SERZZ(L,K)
  300 CONTINUE
      AVG(K)=SUM/15.
      AVG1(K)=SUM1/15.
      AVG2(K)=SUM2/15.

      WRITE(*,1050)AVG(K),AVG1(K),AVG2(K)
  400 CONTINUE
  500 CONTINUE
      GO TO 5

 1000 FORMAT(1X,'MOVING TGT,STAT TGT, OR QUIT(1:MOVE,2:STAT,0:QUIT)')
 1010 FORMAT(1X,'SNRV/35,30,25,20,15,10,7,5,3,1,-2,-5,-7,-10,-20/',/)
 1020 FORMAT(1X,'ENTER SNRV START/STOP POINTS(IE START=1,STOP=15)')
 1025 FORMAT(1X,'PULSES INTEGRATED/1,2,5,10,25,50/',/)
 1027 FORMAT(1X,'ENTER DESIRED NUMBER OF INTEGRATED PULSES')
 1030 FORMAT(/,1X,'SNR LEVEL (DB) = ',E11.5,/)
 1040 FORMAT(14X,'AVGSER',6X,'AVGSERCR',6X,'AVGSERZZ')
 1050 FORMAT(10X,3(3X,E11.5))
 5000 STOP
      END
```

ZIV-ZAKAI BOUND

AVERAGE ERRORS OVER 15 RUNS

OUTPUT RESULTS

FORMAT STATEMENTS IN/OUT

Appendix E

Program Listing

```
*************************************************************************
*     PROGRAM:     SIMULATION OF MULTIPATH EFFECTS (TERRAIN BOUNCE)     *
*     PURPOSE:     COMPUTE THE TRACKING ERROR AND PERFORMANCE BOUNDS    *
*     INPUT:       DESIRED SNR LEVEL(dB), TARGET MOVING OR STATIONARY,  *
*                  AVG OCCURANCES IN THE PRI, SPECULAR OR DIFFUSE MULTI *
*     OUTPUT:      3(1 BY 50) VECTORS CONTAINING AVG ERROR, CRB, ZZB    *
*     COMPILER:    FORTRAN, VERSION 5                                   *
*     EXT CALLS:   IMSL ROUTINES GGNML, MONOR, GGPOS, GGEXN, GGUBS      *
*************************************************************************
      PROGRAM MULPTH                                                    *
                                                                        *  DECLARE VARIABLES
      DIMENSION AVG(50),AVG1(50),AVG2(50),PRE(16),R(1250),B(1250)       *    DIM FOR 24 REFLECT, MAX
      DIMENSION SNRV(15),SER(15,50),SERCR(15,50),SERZZ(15,50)           *
      DIMENSION IR(50),E(26),DTIME(26),U(25),ANG(25)                    *    POIS,EXP,UNIF DEVIATES
      REAL NPWR,N1,N2,NAMP,NCALC,PULSEC                                 *
      INTEGER PULSE,ANGLE,REF,FLAG                                      *    FLAG MONITORS OVERLAP
      DOUBLE PRECISION DSEED1,DSEED2,DSEED3,DSEED4,DSEED5               *    ADDITIONAL DBLEPRE SEEDS
                                                                        *  INPUT STATEMENTS
      DATA SNRV/35.,30.,25.,20.,15.,10.,7.,5.,3.,1.,                    *
     *-2.,-5.,-7.,-10.,-20./                                            *
5     CONTINUE                                                          *
      WRITE(*,1000)                                                     *
      READ*,IMS                                                         *
      IF(IMS.EQ.0)GO TO 5000                                            *    ONLY WAY OUT IS 0 IN IMS
      WRITE(*,1010)                                                     *
      WRITE(*,1020)                                                     *
      READ*,I1,I2                                                       *
                                                                        *  RADAR PARAMETERS
      DATA DSEED1,DSEED2/123457.0D0,325017.0DD/                         *
      DATA DSEED3,DSEED4,DSEED5/123457.0DD,123457.0DD,123457.0DD/       *
      DATA DPRAD,R0,VEL,PW,BW/57.3,4900.,257.4,1.E-6,3.0/               *
      DATA PRI,SIG,OMEGA,AMP,SPWR/10.E-3,636.,636.,5.0,1.E-3/           *
      NR=1250                                                           *    PRIMARY +24 REFLECTED,MAX
      MB=1250                                                           *    PRIMARY +24 REFLECTED,MAX
      NP=50                                                             *    50 POISSON DEVIATES
      FLAG=0                                                            *    INITIALIZE FLAG TO 0
      DO 500 IP=I1,I2                                                   *
      SNR=SNRV(IP)                                                      *
      SNRNEW=10.**(SNR/10.)                                             *
      ROLD=R0                                                           *
      WRITE(*,1023)                                                     *    INPUT AVG OCCURANCES/PRI
      WRITE(*,1025)                                                     *    LAMBDA LIMITED TO 10, POS
      READ*,RLAM                                                        *    RLAM=POISSON PARAMETER
      WRITE(*,1027)                                                     *
      READ*,REF                                                         *    SPECULAR OR DIFFUSE REFL
      IF(RLAM.LE.0.0)GO TO 5                                            *    LOWER LIMIT, MUST BE POS
      IF(RLAM.GT.10.0)GO TO 5                                           *    UPPER LIMIT, AVG ≤ 10
      DO 200 J=1,15                                                     *
      CALL GGNML(DSEED1,NR,R)                                           *
```

```
      CALL GGMNL(DSEED2,NB,B)
      CALL GGPOS(RLAM,DSEED3,NP,IR,IER)

      BSITE=0.0
      TRGT=0.00
      IF(INS.EQ.1)BSITE=1.5
      IF(INS.EQ.1)TRGT=0.0

      THETA=TRGT-BSITE
      I=0
      DO 100 PULSE=1,50
      LAST=IR(PULSE)+1
      NEX=IR(PULSE)
      XM=(1.0E-2)/FLOAT(LAST)
      CALL GGEXN(DSEED4,XM,NEX,E)
      NU=NEX
      CALL GGUBS(DSEED5,NU,U)
      IF(NEX.LE.1)GO TO 65
      DO 60 LL=1,NEX-1
      IP1=LL+1
      DO 55 KK=IP1,NEX
      IF(E(LL).LE.E(KK))GO TO 55
      TEMP=E(LL)
      E(LL)=E(KK)
      E(KK)=TEMP
   55 CONTINUE
   60 CONTINUE

   65 DO 85 MULTI=0,NEX
      IF(FLAG.NE.1)GO TO 67
      FLAG=0
      GO TO 85
   67 CONTINUE
      IF(MULTI.EQ.0)E(MULTI)=0.0
      E(LAST)=PRI
      IF(E(MULTI+1).GE.PRI)GO TO 85
      DTIME(MULTI+1)=E(MULTI+1)-E(MULTI)
      COEFF=0.4
      DO 70 II=1,NU
      ANG(II)=U(II)*6.283
   70 CONTINUE
      IF(MULTI.EQ.0)COEFF=1.0

      IF(DTIME(MULTI+1).GT.2*PW)GO TO 75
      VM1=COEFF
      ANG1=ANG(MULTI)
      VM2=0.4
      ANG2=ANG(MULTI+1)
      VMT=SQRT((VM1*COS(ANG1)+VM2*COS(ANG2))**2
     *+(VM1*SIN(ANG1)+VM2*SIN(ANG2))**2)
      COEFF=VMT
      FLAG=1
   75 CONTINUE
```

```
*                                *
*   POISSON DEVIATES             *
* PLACE TARGET IN BEAMWIDTH      *
*                                *
*                                *
*                                *
*                                *
*                                *
* TRACKING CALCULATIONS          *
*                                *
*                                *
*                                *
*   TOTAL PULSE=POISS+DIRECT     *
*   # REQ INTERARRIVAL TIMES     *
*   COMPUTE AVG = PRI/PULSES     *
*   GENERATE NEX INTRARRIVLS     *
*   MAX # UNFRM DEVIATES REQ     *
*   GENERATE NU ARB PHASES       *
*   DONT NEED TO SORT            *
*   SORT INTERARRIVAL TIMES      *
*                                *
*                                *
*                                *
*                                *
*                                *
*                                *
*                                *
*                                *
*                                *
* MULTIPATH PRI DO LOOP          *
*   DIRECT & REFLCT DO LOOP      *
*   FLAG=1 ON 2ND OVRLP PULSE    *
*   RESET FLAG VARIABLE          *
*   OBSRV 1 PULSE ON OVERLAP     *
*                                *
*   DIRECT PULSE ARRIVAL TIME    *
*   LAST ARRIVAL=NEXT PRI        *
*   LIMIT REF FRAME TO PRI       *
*   DTIME BETWEEN ADJ PULSES     *
*   REF COEFFICIENT SPEC,DIFF    *
*   PHASES UNIFORM 0-2PI         *
*   2PI*U(0,1)                   *
*                                *
*   DIRECT REF COEFF MAG         *
* COMPUTE PULSE OVERLAP          *
*   IF PULSE OVRLP DO ELSE 75    *
*   FIRST PULSE MAG COEFF        *
*   FIRST PULSE PHASE ANGLE      *
*   SECOND PULSE MAG COEFF       *
*   SECOND PULSE PHASE ANGLE     *
*                                *
*   RESULTANT MAG COEFF          *
*                                *
*   FLAG SET OMITS NEXT REF      *
*   END PULSE OVERLAP CALC       *
```

```
      PULSEC=FLOAT(PULSE)                                              *                    *
      TIME=PRI*(PULSEC-1.)+E(MULTI+1)                                  *  CLOCK FOR ENTIRE RUN    *
      RNEW=SQRT(R0**2+(VEL*TIME)**2)                                   *  MOVING TGT RANGE UPDATE *
      IF(IMS.EQ.1.)SNRNEW=((ROLD/RNEW)**4)*SNRNEW                      *                    *
      NPWR=SPWR/SNRNEW                                                 *                    *
      NAMP=SQRT(NPWR)                                                  *                    *
      I=I+1                                                            *                    *
      N1=R(I)*NAMP                                                     *                    *
      N2=B(I)*NAMP                                                     *                    *
      IF(MULTI.EQ.0)GO TO 80                                           *  DIRECT AOA IS TGT LOCAT *
      IF(REF.EQ.1)THETA=U(MULTI)*1.5                                   *  SPEC, AOA IS U(0,BW/2)  *
      IF(REF.EQ.2)THETA=U(MULTI)*180.                                  *  DIFF, AOA IS U(0,PI)    *
      IF(THETA.GT.1.5)GO TO 85                                         *  OMIT ALL REF OUTSIDE BW/2*
80    CONTINUE                                                         *                    *
      EGY=PW*(COEFF*AMP)**2                                            *  EGY IS FUNCTION OF COEFF *
      VD=SIN(1.88*THETA/BW)                                            *                    *
      VS=(SQRT(2.)/2.)*(1.+COS(2.094*THETA/BW))                        *                    *
      R1=COEFF*AMP*SQRT(EGY)*VS+N1                                     *  SUM CHANNEL OBSERVATION *
      R2=COEFF*AMP*SQRT(EGY)*VD+N2                                     *  DIFF CHANNEL OBSERVATION *
      ERR=R1*VD-R2*VS                                                  *                    *
      IF(ERR/(SQRT(2.*EGY)*COEFF*AMP).GE.0.99)                         *                    *
     *ERR=0.99*SQRT(2.*EGY)*COEFF*AMP                                  *                    *
      IF(ERR/(SQRT(2.*EGY)*COEFF*AMP).LE.-0.99)                        *                    *
     *ERR=-0.99*SQRT(2.*EGY)*COEFF*AMP                                 *                    *
      DTHETA=-(BW/1.88)*ASIN(ERR/(SQRT(2.*EGY)*COEFF*AMP))             *                    *
      SERVO=1.0-EXP(-SIG*DTIME(MULTI+1))*(COS(OMEGA*DTIME(MULTI+1))    *                    *
     *-SIN(OMEGA*DTIME(MULTI+1)))                                      *  DTIME IS PULSE TIME DIFF *
      IF(VD*ERR.GT.0.0)DTHETA=-DTHETA                                  *                    *
      BSITE=BSITE+SERVO*DTHETA                                         *  MOVE BORESIGHT EACH PULSE*
85    CONTINUE                                                         *  LOOP END FOR REFLECTIONS *
                                                                       *                    *
      IF(IMS.EQ.1)TRGT=ATAN((VEL*TIME)/R0)*DPRAD                       *  UPDATE MOVE TGT LOCATION *
      THETA=TRGT-BSITE                                                 *  COMPUTE THETA AFTER PRI  *
      BW=BW/DPRAD                                                      *                    *
      THETA=THETA/DPRAD                                                *                    *
                                                                       *  TRACKING ERROR    *
      SER(J,PULSE)=THETA**2                                            *    TRACKING ERROR AFTER PRI *
                                                                       *  COMPUTE CRAMER-RAO BOUND *
      DENOM=(2.*SNRNEW*AMP**2)*(2.865+1.77*COS(3.76*THETA/BW)          *                    *
     *-1.095*COS(4.188*THETA/BW))                                      *                    *
      CRB=(BW**2)/DENOM                                                *                    *
      SERCR(J,PULSE)=CRB                                               *    CRB AFTER PRI   *
                                                                       *  COMPUTE ZIV-ZAKAI BOUND  *
      THETP=(ASIN((SIN(BW/2)+SIN(ABS(THETA)))*0.5))*DPRAD              *                    *
      BW=BW*DPRAD                                                      *                    *
      THETA=THETA*DPRAD                                                *                    *
      DATA PRE/16*0.0/                                                 *                    *
      ANGLE=1.+ABS(INT(10.*THETP))                                     *                    *
      DO 90 N=1,ANGLE                                                  *                    *
      NCALC=(FLOAT(ANGLE)/100.)*N                                      *                    *
      W=5./2+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)            *                    *
     *-COS(3.76*NCALC/BW)                                              *                    *
      RHO=(0.5+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)          *                    *
```

```
     *+COS(3.76*NCALC/BW)/W                                    *                    *
      AX=(SQRT((1.-RHO)*W*SNRNEW/2.))*AMP                      *                    *
      CALL MINOR(AX,PX)                                        *                    *
      QX=1.-PX                                                 *                    *
      NCALC=NCALC/DPRAD                                        *                    *
      PRE(N)=QX*(SIN(1.*NCALC))**2                             *                    *
 90   CONTINUE                                                 *                    *
      DO 94 JK=1,15                                            *                    *
      JK1=JK+1                                                 *                    *
      DO 92 KJ=JK1,16                                          *                    *
      IF(PRE(JK).LE.PRE(KJ))GO TO 92                           *                    *
      STORE=PRE(JK)                                            *                    *
      PRE(JK)=PRE(KJ)                                          *                    *
      PRE(KJ)=STORE                                            *                    *
 92   CONTINUE                                                 *                    *
 94   CONTINUE                                                 *                    *
      ZZB=PRE(16)                                              *                    *
      SERZZ(J,PULSE)=ZZB                                       *  ZZB AFTER PRI     *
                                                               *                    *
 100  CONTINUE                                                 *                    *
 200  CONTINUE                                                 *                    *
      WRITE(*,1030)SNR                                         *                    *
      WRITE(*,1040)                                            *                    *
                                                               *  AVERAGE ERRORS OVER 15 RUNS*
      DO 400 K=1,50                                            *                    *
      SUM=0.0                                                  *                    *
      SUM1=0.0                                                 *                    *
      SUM2=0.0                                                 *                    *
      DO 300 L=1,15                                            *                    *
      SUM=SUM+SER(L,K)                                         *                    *
      SUM1=SUM1+SERCR(L,K)                                     *                    *
      SUM2=SUM2+SERZZ(L,K)                                     *                    *
 300  CONTINUE                                                 *                    *
      AVG(K)=SUM/15.                                           *                    *
      AVG1(K)=SUM1/15.                                         *                    *
      AVG2(K)=SUM2/15.                                         *                    *
                                                               *  OUTPUT RESULTS    *
      WRITE(*,1050)AVG(K),AVG1(K),AVG2(K)                      *                    *
 400  CONTINUE                                                 *                    *
 500  CONTINUE                                                 *                    *
      GO TO 5                                                  *                    *
                                                               *  FORMAT STATEMENTS IN/OUT *
1000 FORMAT(1X,'MOVING TGT,STAT TGT, OR QUIT(1:MOVE,2:STAT,0:QUIT)')  *              *
1010 FORMAT(1X,'SNRV/35,30,25,20,15,10,7,5,3,1,-2,-5,-7,-10,-20/',/)   *              *
1020 FORMAT(1X,'ENTER SNRV START/STOP POINTS(IE START=1,STOP=15)')     *              *
1023 FORMAT(1X,'INPUT RATE OF OCCURANCES/PRI (LAMBDA)')                *              *
1025 FORMAT(1X,'LAMBDA MUST BE REAL (0-10.0)',/)                       *              *
1027 FORMAT(/,1X,'SPECULAR OR DIFFUSE REF?(1:SPEC,2:DIFF)',/)          *              *
1030 FORMAT(/,1X,'SNR LEVEL (DB) = ',E11.5,/)                          *              *
1040 FORMAT(14X,'AVGSER',8X,'AVGSERCR',6X,'AVGSERZZ')                  *              *
1050 FORMAT(10X,3(3X,E11.5))                                           *              *
5000 STOP                                                              *              *
      END                                                              *              *
```

Appendix F

Program Listing

```
*************************************************************************
*     PROGRAM:    "LOCKOUT" OF UNDESIRED PULSES WITHIN PRI(REQ 4 PULSES)*
*     PURPOSE:    COMPUTE THE TRACKING ERROR AND PERFORMANCE BOUNDS     *
*     INPUT:      DESIRED SNR LEVEL(dB), TARGET MOVING OR STATIONARY,   *
*                 AVG OCCURANCES IN THE PRI, SPECULAR OR DIFFUSE MULTI  *
*     OUTPUT:     3(1 BY 50) VECTORS CONTAINING AVG ERROR, CRB, ZZB     *
*     COMPILER:   FORTRAN, VERSION 5                                    *
*     EXT CALLS:  IMSL ROUTINES GGNML, MDNOR, GGPOS, GGEXN, GGUBS       *
*************************************************************************
      PROGRAM MULFIX                                                    *
                                                                        *
                                                  * DECLARE VARIABLES    *
      DIMENSION AVG(50),AVG1(50),AVG2(50),PRE(16),R(1250),B(1250)       *
      DIMENSION SNRV(15),SER(15,50),SERCR(15,50),SERZZ(15,50)           *
      DIMENSION IR(50),E(26),DTIME(26),U(25),ANG(25),ARVAL(4,26)        *  DIM ARRIVAL TIMES 4 PRI'S*
      REAL NPWR,N1,N2,NAMP,NCALC,PULSEC                                 *
      INTEGER PULSE,ANGLE,REF,FLAG,FLAG1,FLAG2,FLAG3                    *  THREE ADDITIONAL FLAGS  *
      DOUBLE PRECISION DSEED1,DSEED2,DSEED3,DSEED4,DSEED5               *
                                                  * INPUT STATEMENTS     *
      DATA SNRV/35.,30.,25.,20.,15.,10.,7.,5.,3.,1.,                    *
     *-2.,-5.,-7.,-10.,-20./                                            *
 5    CONTINUE                                                          *
      WRITE(*,1000)                                                     *
      READ*,IMS                                                         *
      IF(IMS.EQ.0)GO TO 5000                                            *  ONLY WAY OUT IS 0 IN IMS *
      WRITE(*,1010)                                                     *
      WRITE(*,1020)                                                     *
      READ*,I1,I2                                                       *
                                                  * RADAR PARAMETERS     *
      DATA DSEED1,DSEED2/123457.000,325017.000/                         *
      DATA DSEED3,DSEED4,DSEED5/123457.000,123457.000,123457.000/       *
      DATA DPRAD,R0,VEL,PW,BW/57.3,4900.,257.4,1.E-6,3.0/               *
      DATA PRI,SIG,OMEGA,AMP,SPWR/10.E-3,636.,636.,5.0,1.E-3/           *
      NR=1250                                                           *
      NB=1250                                                           *
      NP=50                                                             *
      FLAG=0                                                            *
      DO 500 IP=I1,I2                                                   *
      SNR=SNRV(IP)                                                      *
      SNRNEW=10.**(SNR/10.)                                             *
      ROLD=R0                                                           *
      WRITE(*,1023)                                                     *
      WRITE(*,1025)                                                     *
      READ*,RLAM                                                        *
      WRITE(*,1027)                                                     *
      READ*,REF                                                         *
      IF(RLAM.LE.0.0)GO TO 5                                            *
      IF(RLAM.GT.10.0)GO TO 5                                           *
      DO 200 J=1,15                                                     *
      DATA ARVAL/104*1.0/                                               *  INITIALIZE ALL ARVAL=1.0 *
```

```
      FLAG3=0                                                    *   INIT FLAG3 (LOCK INDICAT)*
      CALL GGNML(DSEED1,NR,R)                                    *                            *
      CALL GGNML(DSEED2,NB,B)                                    *                            *
      CALL GGPOS(RLAM,DSEED3,NP,IR,IER)                          *                            *
                                                                 * PLACE TARGET IN BEAMWIDTH  *
      BSITE=0.0                                                  *                            *
      TRGT=0.80                                                  *                            *
      IF(IMS.EQ.1)BSITE=1.5                                      *                            *
      IF(IMS.EQ.1)TRGT=0.0                                       *                            *
                                                                 * TRACKING CALCULATIONS       *
      THETA=TRGT-BSITE                                           *                            *
      I=0                                                        *                            *
      DO 100 PULSE=1,50                                          *                            *
      LAST=IR(PULSE)+1                                           *                            *
      NEX=IR(PULSE)                                              *                            *
      XM=(1.0E-2)/FLOAT(LAST)                                    *                            *
      CALL GGEXN(DSEED4,XM,NEX,E)                                *                            *
      NU=NEX                                                     *                            *
      CALL GGUBS(DSEED5,NU,U)                                    *                            *
      IF(NEX.LE.1)GO TO 65                                       *                            *
      DO 60 LL=1,NEX-1                                           *                            *
      IP1=LL+1                                                   *                            *
      DO 55 KK=IP1,NEX                                           *                            *
      IF(E(LL).LE.E(KK))GO TO 55                                 *                            *
      TEMP=E(LL)                                                 *                            *
      E(LL)=E(KK)                                                *                            *
      E(KK)=TEMP                                                 *                            *
55    CONTINUE                                                   *                            *
60    CONTINUE                                                   *                            *
                                                                 * MULTIPATH PRI DO LOOP       *
65    DO 85 MULTI=0,NEX                                          *                            *
      IF(FLAG.NE.1)GO TO 67                                      *                            *
      FLAG=0                                                     *                            *
      GO TO 85                                                   *                            *
67    CONTINUE                                                   *                            *
      IF(MULTI.EQ.0)E(MULTI)=0.0                                 *                            *
      E(LAST)=PRI                                                *                            *
      IF(E(MULTI+1).GE.PRI)GO TO 85                              *                            *
      DTIME(MULTI+1)=E(MULTI+1)-E(MULTI)                         *                            *
      IF(FLAG3.EQ.1)DTIME(MULTI+1)=PRI                           *   ON LOCK OBS PERIOD IS PRI*
      COEFF=0.4                                                  *                            *
      DO 70 II=1,NU                                              *                            *
      ANG(II)=U(II)*6.283                                        *                            *
70    CONTINUE                                                   *                            *
      IF(MULTI.EQ.0)COEFF=1.0                                    *                            *
                                                                 * COMPUTE PULSE OVERLAP       *
      IF(DTIME(MULTI+1).GT.2*PW)GO TO 75                         *                            *
      VM1=COEFF                                                  *                            *
      ANG1=ANG(MULTI)                                            *                            *
      VM2=0.4                                                    *                            *
      ANG2=ANG(MULTI+1)                                          *                            *
      VMT=SQRT((VM1*COS(ANG1)+VM2*COS(ANG2))**2                  *                            *
     *+(VM1*SIN(ANG1)+VM2*SIN(ANG2))**2)                         *                            *
```

```
      COEFF=AMT                                                        *                          *
      FLAG=1                                                           *   PULSE OVERLAP MONITOR  *
   75 CONTINUE                                                         *                          *
                                                                       *   END PULSE OVERLAP COMPUTE*
      PULSEC=FLOAT(PULSE)                                              *                          *
      TIME=PRI*(PULSEC-1.)+E(MULTI+1)                                  *                          *
      RNEW=SQRT(R0**2+(VEL*TIME)**2)                                   *                          *
      IF(IMS.EQ.1.)SNRNEW=((ROLD/RNEW)**4)*SNRNEW                      *                          *
      NPWR=SPWR/SNRNEW                                                 *                          *
      NAMP=SQRT(NPWR)                                                  *                          *
      I=I+1                                                            *                          *
      N1=R(I)*NAMP                                                     *                          *
      N2=B(I)*NAMP                                                     *                          *
      IF(MULTI.EQ.0)GO TO 80                                           *                          *
      IF(REF.EQ.1)THETA=U(MULTI)*1.5                                   *                          *
      IF(REF.EQ.2)THETA=U(MULTI)*180.                                  *                          *
      IF(THETA.GT.1.5)GO TO 85                                         *                          *
   80 CONTINUE                                                         *                          *
                                                                       *  COMPUTE PULSE "LOCK"    *
      IF(PULSE.LE.4)ARVAL(PULSE,MULTI+1)=E(MULTI)                      *    STORE ARVAL TIMES FIRST 4*
      IF(PULSE.NE.5)GO TO 84                                           *    LOOK FOR PERIODIC ON 5TH *
      DO 82 JJJ=1,25                                                   *    CONSIDER 25 PULSES POSSIB*
      TOTAL=(ARVAL(1,JJJ)+ARVAL(2,JJJ)+ARVAL(3,JJJ)+ARVAL(4,JJJ))/4.0  *    AVG 25 PULSES OVR 4 PRI'S*
      IF(TOTAL.NE.ARVAL(1,JJJ))GO TO 82                                *    DOES AVG=PRI1,PULSE JJJ? *
      IF(TOTAL.NE.ARVAL(2,JJJ))GO TO 82                                *    DOES AVG=PRI2,PULSE JJJ? *
      IF(TOTAL.NE.ARVAL(3,JJJ))GO TO 82                                *    DOES AVG=PRI3,PULSE JJJ? *
      IF(TOTAL.NE.ARVAL(4,JJJ))GO TO 82                                *    DOES AVG=PRI4,PULSE JJJ? *
      IF(TOTAL.EQ.1.0)GO TO 82                                         *    RECALL ARVAL INIT TO 1.0 *
      IF(TOTAL.EQ.PRI)GO TO 82                                         *    NEXT PRI NOT DESIRED     *
      FLAG2=JJJ-1                                                      *    JJJ-1 BECAUSE DIRECT IS 0*
      FLAG3=1                                                          *    SET & LOCKED TO PULSE FL2*
   82 CONTINUE                                                         *    LOOP THRU ALL 25 PULSES  *
   84 CONTINUE                                                         *    OUT IF ROW,COL .NE. AVG  *
                                                                       *    END PULSE LOCK COMPUTE   *
      EGY=PW*(COEFF*AMP)**2                                            *                          *
      VO=SIN(1.88*THETA/BW)                                            *                          *
      VS=(SQRT(2.)/2.)*(1.+COS(2.894*THETA/BW))                        *                          *
      R1=COEFF*AMP*SQRT(EGY)*VS+N1                                     *                          *
      R2=COEFF*AMP*SQRT(EGY)*VO+N2                                     *                          *
      ERR=R1*VO-R2*VS                                                  *                          *
      IF(ERR/(SQRT(2.*EGY)*COEFF*AMP).GE.0.99)                         *                          *
     *ERR=0.99*SQRT(2.*EGY)*COEFF*AMP                                  *                          *
      IF(ERR/(SQRT(2.*EGY)*COEFF*AMP).LE.-0.99)                        *                          *
     *ERR=-0.99*SQRT(2.*EGY)*COEFF*AMP                                 *                          *
      DTHETA=-(BW/1.88)*ASIN(ERR/(SQRT(2.*EGY)*COEFF*AMP))             *                          *
      SERVO=1.0-EXP(-SIG*DTIME(MULTI+1))*(COS(OMEGA*DTIME(MULTI+1))    *                          *
     *-SIN(OMEGA*DTIME(MULTI+1)))                                      *                          *
      IF(VO*ERR.GT.0.0)DTHETA=-DTHETA                                  *                          *
      IF(MULTI.EQ.FLAG2)BSITE=BSITE+SERVO*DTHETA                       *  LOCK ON FLAG2, OMIT OTHER*
   85 CONTINUE                                                         *  END MULTIPATH DO LOOP    *
                                                                       *                           *
      IF(IMS.EQ.1)TRGT=ATAN(VEL*TIME)/R0)*DPRAD                        *                          *
      THETA=TRGT-BSITE                                                 *                          *
```

```
      BW=BW/DPRAD
      THETA=THETA/DPRAD

      SER(J,PULSE)=THETA**2

      DENOM=(2.*SNRNEW*AMP**2)*(2.865+1.77*COS(3.76*THETA/BW)
     *-1.895*COS(4.188*THETA/BW))
      CRB=(BW**2)/DENOM
      SERCR(J,PULSE)=CRB

      THETP=(ASIN((SIN(BW/2)+SIN(ABS(THETA)))*0.5))*DPRAD
      BW=BW*DPRAD
      THETA=THETA*DPRAD
      DATA PRE/16*0.0/
      ANGLE=1.+ABS(INT(10.*THETP))
      DO 90 N=1,ANGLE
      NCALC=(FLOAT(ANGLE)/100.)*N
      W=5./2+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)
     *-COS(3.76*NCALC/BW)
      RHO=(0.5+2.*COS(2.094*NCALC/BW)+0.5*COS(4.188*NCALC/BW)
     *+COS(3.76*NCALC/BW))/W
      AX=(SQRT((1.-RHO)*W*SNRNEW/2.))*AMP
      CALL MINOR(AX,PX)
      QX=1.-PX
      NCALC=NCALC/DPRAD
      PRE(N)=QX*(SIN(1.*NCALC))**2
   90 CONTINUE
      DO 94 JK=1,15
      JK1=JK+1
      DO 92 KJ=JK1,16
      IF(PRE(JK).LE.PRE(KJ))GO TO 92
      STORE=PRE(JK)
      PRE(JK)=PRE(KJ)
      PRE(KJ)=STORE
   92 CONTINUE
   94 CONTINUE
      ZZB=PRE(16)
      SERZZ(J,PULSE)=ZZB

  100 CONTINUE
  200 CONTINUE
      WRITE(*,1030)SNR
      WRITE(*,1040)

      DO 400 K=1,50
      SUM=0.0
      SUM1=0.0
      SUM2=0.0
      DO 300 L=1,15
      SUM=SUM+SER(L,K)
      SUM1=SUM1+SERCR(L,K)
      SUM2=SUM2+SERZZ(L,K)
  300 CONTINUE
```

TRACKING ERROR

COMPUTE CRAMER-RAO BOUND

COMPUTE ZIV-ZAKAI BOUND

AVERAGE ERRORS OVER 15 RUNS

```
      AVG(K)=SUM/15.                                                    *                    *
      AVG1(K)=SUM1/15.                                                  *                    *
      AVG2(K)=SUM2/15.                                                  *                    *
                                                                        *  OUTPUT RESULTS    *
      WRITE(*,1050)AVG(K),AVG1(K),AVG2(K)                               *                    *
  400 CONTINUE                                                          *                    *
  500 CONTINUE                                                          *                    *
      GO TO 5                                                           *                    *
                                                                        *  FORMAT STATEMENTS IN/OUT *
 1000 FORMAT(1X,'MOVING TGT,STAT TGT, OR QUIT(1:MOVE,2:STAT,0:QUIT)')   *                    *
 1019 FORMAT(1X,'SNRV/35,30,25,20,15,10,7,5,3,1,-2,-5,-7,-10,-20/',/)    *                    *
 1020 FORMAT(1X,'ENTER SNRV START/STOP POINTS(IE START=1,STOP=15)')      *                    *
 1023 FORMAT(1X,'INPUT RATE OF OCCURANCES/PRI (LAMBDA)')                 *                    *
 1025 FORMAT(1X,'LAMBDA MUST BE REAL (0-<0.8)',/)                        *                    *
 1027 FORMAT(/,1X,'SPECULAR OR DIFFUSE REF?(1:SPEC,2:DIFF)',/)           *                    *
 1030 FORMAT(/,1X,'SNR LEVEL (DB) = ',E11.5,/)                           *                    *
 1040 FORMAT(14X,'AVGSER',8X,'AVGSERCR',6X,'AVGSERZZ')                   *                    *
 1050 FORMAT(10X,3(3X,E11.5))                                            *                    *
 5000 STOP                                                              *                    *
      END                                                               *                    *
```

Appendix G

Program Listing

```
*************************************************************************
*     PROGRAM:     TWO-DIMENSIONAL AC MONOPULSE RADAR SIMULATION        *
*     PURPOSE:     COMPUTE THE TRACKING ERROR AND PERFORMANCE BOUND     *
*     INPUT:       DESIRED SNR LEVEL(dB)                               *
*     OUTPUT:      2(1 BY 50) VECTORS CONTAINING AVG ERROR & ZZB        *
*     COMPILER:    FORTRAN, VERSION 5                                  *
*     EXT CALLS:   IMSL ROUTINES GGNML & MDNOR                         *
*************************************************************************
      PROGRAM TWODIM                                                   *
                                                                       *
                                                  DECLARE VARIABLES    *
      DIMENSION AVG(50),AVG2(50),PRE(16,16),R(50),B(50),C(50)          *
      DIMENSION SNRV(15),SER(15,50),SER22(15,50)                       *
      REAL NPWR,N1,N2,N3,NAMP,NCALC1,NCALC2                            *
      INTEGER PULSE,ANGLE1,ANGLE2                                      *
      DOUBLE PRECISION DSEED1,DSEED2,DSEED3                            *
      DATA SNRV/35.,30.,25.,20.,15.,10.,7.,5.,3.,1.,                   *
     *-2.,-5.,-7.,-10.,-20./                                           *
    5 CONTINUE                                                         *
      WRITE(*,1000)                                                    *
      READ*,IMS                                                        *
      IF(IMS.EQ.0)GO TO 5000                                           *
      WRITE(*,1010)                                                    *
      WRITE(*,1020)                                                    *
      READ*,I1,I2                                                      *
                                                  RADAR PARAMETERS     *
      DATA DSEED1,DSEED2,DSEED3/123457.0D0,325017.0D0,133547.0D0/      *
      DATA DPRAD,PW,BW/57.3,1.E-6,3.0/                                 *
      DATA PRI,SIG,OMEGA,AMP,SPWR/10.E-3,636.,636.,5.0,1.E-3/          *
      EGY=PW*AMP**2                                                    *
      NR=50                                                            *
      NB=50                                                            *
      NC=50                                                            *
      DO 500 IP=I1,I2                                                  *
      SNR=SNRV(IP)                                                     *
      SNRNEW=10.**(SNR/10.)                                            *
      DO 200 J=1,15                                                    *
      CALL GGNML(DSEED1,NR,R)                                          *
      CALL GGNML(DSEED2,NB,B)                                          *
      CALL GGNML(DSEED3,NC,C)                                          *
                                                  PLACE TARGET IN BEAMWIDTH *
      BSITEL=0.0                                                       *
      BSITEA=0.0                                                       *
      TRGTEL=0.80                                                      *
      TRGTAZ=0.60                                                      *
                                                  TRACKING CALCULATIONS *
      THETA=TRGTEL-BSITEL                                              *
      PHI=TRGTAZ-BSITEA                                                *
      I=0                                                              *
      DO 100 PULSE=1,50                                                *
```

```
PULSEC=FLOAT(PULSE)                                                    *                *
NPWR=SPWR/SNRNEW                                                       *                *
NAMP=SQRT(NPWR)                                                        *                *
I=I+1                                                                  *                *
N1=R(I)*NAMP                                                           *                *
N2=B(I)*NAMP                                                           *                *
N3=C(I)*NAMP                                                           *                *
VDEL=THETA/BW                                                          *                *
VDAZ=PHI/BW                                                            *                *
VS=1.                                                                  *                *
R1=AMP*SQRT(EGY)*VS+N1                                                 *                *
R2=AMP*SQRT(EGY)*VDEL+N2                                               *                *
R3=AMP*SQRT(EGY)*VDAZ+N3                                               *                *
ERREL=R1*BW*THETA+R3*THETA*PHI-R2*(BW**2+PHI**2)                       *                *
ERRAZ=R1*BW*PHI+R2*THETA*PHI-R3*(BW**2+THETA**2)                       *                *
DTHETA=-ERREL/(AMP*SQRT(EGY)*BW)                                       *                *
DPHI=-ERRAZ/(AMP*SQRT(EGY)*BW)                                         *                *
SERVEL=1.0-EXP(-SIG*PRI)*(COS(OMEGA*PRI)-SIN(OMEGA*PRI))               *                *
SERVAZ=1.0-EXP(-SIG*PRI)*(COS(OMEGA*PRI)-SIN(OMEGA*PRI))               *                *
IF(VDEL*ERREL.GT.0.0)DTHETA=-DTHETA                                    *                *
IF(VDAZ*ERRAZ.GT.0.0)DPHI=-DPHI                                        *                *
BSITEL=BSITEL+SERVEL*DTHETA                                            *                *
BSITEA=BSITEA+SERVAZ*DPHI                                              *                *
THETA=TRGTEL-BSITEL                                                    *                *
PHI=TRGTAZ-BSITEA                                                      *                *
BW=BW/DPRAD                                                            *                *
THETA=THETA/DPRAD                                                      *                *
PHI=PHI/DPRAD                                                          *                *
                                                                       *  TRACKING ERROR & BOUNDS  *
SER(J,PULSE)=(THETA**2+PHI**2)                                         *    TRACKING ERROR      *
THETP=(ASIN((SIN(BW/2)+SIN(ABS(THETA)))*0.5))*DPRAD                    *                *
PHIP=(ASIN((SIN(BW/2)+SIN(ABS(PHI)))*0.5))*DPRAD                       *                *
BW=BW*DPRAD                                                            *                *
THETA=THETA*DPRAD                                                      *                *
PHI=PHI*DPRAD                                                          *                *
DATA PRE/256*0.0/                                                      *                *
ANGLE1=1.+ABS(INT(10.*THETP))                                          *                *
ANGLE2=1.+ABS(INT(10.*PHIP))                                           *                *
DO 90 N=1,ANGLE1                                                       *                *
DO 80 NN=1,ANGLE2                                                      *                *
NCALC1=(FLOAT(ANGLE1)/100.)*N                                          *                *
NCALC2=(FLOAT(ANGLE2)/100.)*NN                                         *                *
W1=5./2+2.*COS(2.094*NCALC1/BW)+0.5*COS(4.188*NCALC1/BW)               *                *
*-COS(3.76*NCALC1/BW)                                                  *                *
W2=5./2+2.*COS(2.094*NCALC2/BW)+0.5*COS(4.188*NCALC2/BW)               *                *
*-COS(3.76*NCALC2/BW)                                                  *                *
RHO1=(0.5+2.*COS(2.094*NCALC1/BW)+0.5*COS(4.188*NCALC1/BW)             *                *
*+COS(3.76*NCALC1/BW)/W1                                               *                *
RHO2=(0.5+2.*COS(2.094*NCALC2/BW)+0.5*COS(4.188*NCALC2/BW)             *                *
*+COS(3.76*NCALC2/BW)/W2                                               *                *
AX1=(SQRT((1.-RHO1)*W1*(SNRNEW/2.)))*AMP                               *                *
AX2=(SQRT((1.-RHO2)*W2*(SNRNEW/2.)))*AMP                               *                *
AX31=(AMP*NCALC1/BW)*SQRT(2.*SNRNEW)                                   *                *
```

```
      ST=SQRT((NCALC1/DPRAD)**2+(NCALC2/DPRAD)**2)                    *                    *
      AX32=(AMP*ST*DPRAD/BW)*SQRT(2.*SNRNEW)                          *                    *
      AX33=(AMP*NCALC2/BW)*SQRT(2.*SNRNEW)                            *                    *
      CALL MDNOR(AX1,PX1)                                             *                    *
      CALL MDNOR(AX2,PX2)                                             *                    *
      CALL MDNOR(AX31,PX31)                                           *                    *
      CALL MDNOR(AX32,PX32)                                           *                    *
      CALL MDNOR(AX33,PX33)                                           *                    *
      QX1=1.-PX1                                                      *                    *
      QX2=1.-PX2                                                      *                    *
      QX31=1.-PX31                                                    *                    *
      QX32=1.-PX32                                                    *                    *
      QX33=1.-PX33                                                    *                    *
      NCALC1=NCALC1/DPRAD                                             *                    *
      NCALC2=NCALC2/DPRAD                                             *                    *
      PEST=.333*(QX31+QX32+QX33)                                      *                    *
      PRE(N,NN)=.125*(SIN(ST)**2)*PEST+(2./(SIN(ST)**2))              *                    *
     +*(QX1*SIN(1.*NCALC1)**2)*(QX2*SIN(1.*NCALC2)**2)                *                    *
      NCALC1=NCALC1*DPRAD                                             *                    *
      NCALC2=NCALC2*DPRAD                                             *                    *
 80   CONTINUE                                                        *                    *
 90   CONTINUE                                                        *                    *
      DO 96 JJ=1,15                                                   *                    *
      DO 94 JK=1,15                                                   *                    *
      JK1=JK+1                                                        *                    *
      DO 92 KJ=JK1,16                                                 *                    *
      IF(PRE(JJ,JK).LE.PRE(JJ,KJ))GO TO 92                           *                    *
      STORE1=PRE(JJ,JK)                                               *                    *
      PRE(JJ,JK)=PRE(JJ,KJ)                                           *                    *
      PRE(JJ,KJ)=STORE1                                               *                    *
 92   CONTINUE                                                        *                    *
 94   CONTINUE                                                        *                    *
 96   CONTINUE                                                        *                    *
      DO 98 JL=1,15                                                   *                    *
      JL1=JL+1                                                        *                    *
      DO 97 LJ=JL1,16                                                 *                    *
      IF(PRE(JL,16).LE.PRE(LJ,16))GO TO 97                            *                    *
      STORE2=PRE(JL,16)                                               *                    *
      PRE(JL,16)=PRE(LJ,16)                                           *                    *
      PRE(LJ,16)=STORE2                                               *                    *
 97   CONTINUE                                                        *                    *
 98   CONTINUE                                                        *                    *
      ZZB=PRE(16,16)                                                  *  ZIV-ZAKAI BOUND   *
      SERZZ(J,PULSE)=ZZB                                              *                    *
100   CONTINUE                                                        *                    *
200   CONTINUE                                                        *                    *
      WRITE(*,1030)SNR                                                *                    *
      WRITE(*,1040)                                                   *                    *
                                                                      *  AVERAGE ERRORS OVER 15 RUNS*
      DO 400 K=1,50                                                   *                    *
      SUM=0.0                                                         *                    *
      SUM2=0.0                                                        *                    *
      DO 300 L=1,15                                                   *                    *
```

```
      SUM=SUM+SER(L,K)                                          #                    #
      SUM2=SUM2+SERZZ(L,K)                                      #                    #
  300 CONTINUE                                                  #                    #
      AVG(K)=SUM/15.                                            #                    #
      AVG2(K)=SUM2/15.                                          #                    #
                                                                #  OUTPUT RESULTS    #
      WRITE(*,1050)AVG(K),AVG2(K)                               #                    #
  400 CONTINUE                                                  #                    #
  500 CONTINUE                                                  #                    #
      GO TO 5                                                   #                    #
                                                                #  FORMAT STATEMENTS IN/OUT #
 1000 FORMAT(1X,'QUIT?(ENTER 1 TO PROCEED, 0 TO QUIT)')         #                    #
 1010 FORMAT(1X,'SNRV/35,30,25,20,15,10,7,5,3,1,-2,-5,-7,-10,-20/',/)  #             #
 1020 FORMAT(1X,'ENTER SNRV START/STOP POINTS(IE START=1,STOP=15)')    #             #
 1030 FORMAT(/,1X,'SNR LEVEL (DB) = ',E11.5,/)                  #                    #
 1040 FORMAT(14X,'AVGSER',25X,'AVGSERZZ')                       #                    #
 1050 FORMAT(10X,3X,E11.5,17X,E11.5)                            #                    #
 5000 STOP                                                      #                    #
      END                                                       #                    #
```

Bibliography

1. Stutzman, W.L., and G.A. Thiele. Antenna Theory and Design. New York: John Wiley & Sons, Inc., 1981.

2. Papoulis, A. The Fourier Integral and its Applications. New York: McGraw-Hill Book Company, 1962.

3. Peebles, P.Z., Jr. Communication System Principles. (Fourth Printing). Massachusetts: Addison-Wesley Publishing Company, 1976.

4. Golden, Major August, Jr. Radar Electronic Warfare. Lecture materials distributed in EE 5.73, Electronic Warfare. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, May 1983.

5. Van Trees, H.L. Detection, Estimation, and Modulation Theory. (Part I). New York: John Wiley & Sons, Inc., 1968.

6. Wozencraft, J.M., and I.M. Jacobs. Principles of Communication Engineering. New York: John Wiley & Sons, Inc., 1965.

7. Soong, T.T. Probabilistic Modeling and Analysis in Science and Engineering. New York: John Wiley & Sons, Inc., 1981.

8. Melsa, J.L., and D.L. Cohn. Decision and Estimation Theory. New York: McGraw-Hill Book Company, 1978.

9. Seidman, L.P. "Bearing Estimation Error with a Linear Array," IEEE Transactions on Audio and Electroacoustics, AU-19, No. 2: 147-157, (June 1971).

10. Ziv, J., and M. Zakai. "Some Lower Bounds on Signal Parameter Estimation," IEEE Transactions on Information Theory, IT-15, No. 3: 386-391, (May 1969).

11. Papoulis, A. Probability, Random Variables and Stochastic Processes. New York: McGraw-Hill Book Company, 1965.

12. Seidman, L.P. "Performance Limitations and Error Calculations for Parameter Estimation," Proceedings of the IEEE, Vol. 58, No. 5: 644-652, (May 1970).

13. Shinners, S.M. Control System Synthesis. New York: John Wiley & Sons, Inc., 1964.

14. Locke, A.S. Guidance. New Jersey: D. Van Nostrand Company, Inc., 1955.

15. Skolnik, M.I. Introduction to Radar Systems. (Second Edition). New York: McGraw-Hill Book Company, 1980.

16. Barton, D.K. "Low-Angle Radar Tracking," Proceedings of the IEEE, Vol. 62, No. 6: 687-704, (June 1974).

17. Proakis, J.G. Digital Communications. New York: McGraw-Hill Book Company, 1983.

18. DiFranco J.V. and W.L. Rubin. Radar Detection. New Jersey: Prentice-Hall, Inc., 1968.

19. McGarty, T.P. Models of Multipath Propagation effects in an Air to Ground Surveillance System. M.I.T. Lincoln Laboratory, Lexington, Mass., Report ATC-24, 1974. (AD 777 241)

20. Develet, J.A. "Thermal-Noise Errors in Simultaneous-Lobing and Conical-Scan Angle-Tracking Systems," IRE Transactions on Space Electronics and Telemetry, Vol. 7, No. 2: 42-51, (June 1961).

21. Urkowitz, H. "The Accuracy of Maximum Likelihood Angle Estimates in Radar and Sonar," IEEE Transactions on Military Electronics, Vol. 8, No. 1: 39-45, (January 1964).

22. Parzen, E. Stochastic Processes. San Francisco: Holden-Day, Inc., 1962.

VITA

Terry E. Smith was born on 6 November 1953 in Mt. Gilead, Ohio. He graduated from high school in Cardington, Ohio in 1971. He entered the Air Force in 1973 and served as a Radio Relay Equipment Repairman. While on active duty, he attended Allan Hancock College, Santa Maria, California from which he received the degree of Associate in Arts in Liberal Arts in June 1977. Upon completion of his four year active duty obligation, he both attended California Polytechnic State University, San Luis Obispo, California and was employed as an Engineering Student for Hughes Aircraft Company, El Segundo, California on a cooperative education program. He graduated with honors with the degree of Bachelor of Science in Electronic Engineering in June 1982. Upon graduation, he was employed as a Standards Engineer for Sperry Univac Company, Salt Lake City, Utah. He entered the Air Force on active duty in January 1983, and received his commission from Officer Training School in April 1983. His first assignment was to the School of Engineering, Air Force Institute of Technology, in June 1983.

Permanent address: 7326 Candlewood Lake
Mt. Gilead, Ohio 43338

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/84D-62	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification): See Box 19	
12. PERSONAL AUTHOR(S): Terry E. Smith, B.S., 2d Lt, USAF	
13a. TYPE OF REPORT: MS Thesis; 13b. TIME COVERED: FROM _____ TO _____; 14. DATE OF REPORT (Yr., Mo., Day): 1984 December; 15. PAGE COUNT: 171	
16. SUPPLEMENTARY NOTATION: (19)	
17. COSATI CODES: FIELD 17, GROUP 09, SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Monopulse Radar, Radar Tracking, Angle Estimation, Ziv-Zakai Bound, Target Fluctuations, Multipath

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: ZIV-ZAKAI BOUND APPLIED TO AN AMPLITUDE-COMPARISON MONOPULSE RADAR

Thesis Chairman: Ronald J. Carpinella, Lt. Col, USAF

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 25 Feb 85
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Ronald J. Carpinella, Lt. Col, USAF	22b. TELEPHONE NUMBER (Include Area Code): 513-257-6126; 22c. OFFICE SYMBOL: FTD/TQC

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

This investigation estimates the target angle off-boresight of a monopulse radar, generates an error signal, and excites a servo system to reposition the boresight on target, thus providing target tracking. Cramer-Rao (CR) and Ziv-Zakai (ZZ) bounds are derived to indicate system performance under varying signal-to-noise ratio (SNR) conditions. Actual tracking error is approximated from a computer simulation and compared to the CR and ZZ bounds for varying SNR levels, for tracking in "slow" and "rapid" target fluctuation environments, and for tracking in the presence of specular and diffuse multipath. A two-dimensional tracking model and associated performance bound are also presented.

At high SNR levels, the CR bound results lower bounded the mean-square tracking error (MSE), but for low SNR conditions the CR bound results exceeded the MSE. The ZZ bound results indicate a tight lower bound for the MSE at low SNR levels and in both the multipath and target fluctuation environments. "Slow" target fluctuations and diffuse multipath results indicate that target tracking capability is not seriously degraded in either of these two surroundings. Conversely, "Rapid" target fluctuations and specular multipath environment results indicate serious tracking degradation is introduced in the monopulse tracker.

ORIGINATOR - SUPPLIED KEY WORDS INCLUDE:

END

FILMED

4-85

DTIC